A CO-EVOLUTIONARY MULTI-AGENT APPROACH FOR DESIGNING THE ARCHITECTURE OF RECONFIGURABLE MANUFACTURING MACHINES

A Thesis
Presented to
the Academic Faculty

By

Nathan P. Young

In Partial Fulfillment
of the Requirements for The Degree
Master of Science in Mechanical Engineering in the
School of George W. Woodruff School of Mechanical Engineering

Georgia Institute of Technology
August 2008

A CO-EVOLUTIONARY MULTI-AGENT APPROACH FOR DESIGNING THE ARCHITECTURE OF RECONFIGURABLE MANUFACTURING MACHINES

Approved by:

Dr. Mervyn Fathianathan, Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Shreyes Melkote
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Chris Paredis
School of Mechanical Engineering
Georgia Institute of Technology

Date Approved: April 21st, 2008

ACKNOWLEDGEMENTS

During the completion of this thesis, I received a great deal of intellectual guidance and encouragement from many people.

First and foremost I acknowledge my advisor, Mervyn Fathianathan, for his dedication to my professional development. From my first days in the Systems Realization Laboratory, Mervyn helped me to realize my true intellectual potential and achieve goals I never thought possible. Also, I would like to thank my reading committee Chris Paredis and Shreyes Melkote. While working with Chris, I published my first conference paper and gained great insight into the discipline of modeling and simulation. I am also grateful to Shreyes Melkote for reading my thesis and providing expertise in manufacturing.

Also, I would like to extend my gratitude to the students of the SRL. First of all, I thank my unofficial thesis advisors Matthew Chamberlain, Chris Williams, Matthias Messer, and Stephanie Thompson. Matthew, Chris, Matthias, and Stephanie gave encouragement and advice during the final months of my thesis. I thank many SRL students for their friendship and help in completing this thesis, including Alex Ruderman, Jonathan Bankston, Jonathan Jobe, Jay Johnson, Kenway Chen, Tommy Johnson, Hannah Muchnick, Jitesh Panchal, Jamal Wilson, John Reap, Hao Wu, and Justin Rockwell.

I would like to thank my family for their constant love and support. My parents offered encouraging words in times of difficulty. From them, I learned the importance of

perseverance, and for that I am thankful. I am also grateful to my siblings for offering kind words and help during this work.

Finally, I am especially grateful for my fiancée, Sarah Langguth. Throughout my graduate studies, she has listened to my concerns and worries with the utmost care. Her selfless dedication and sound advice provided me with comfort and guidance when I needed it most. It is through her constant love and friendship; I found the motivation and strength to complete this work.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Manufacturing companies today face increasingly uncertain and volatile market demands. Product designs and the required quantities change rapidly to meet the needs of customers. To maintain competitiveness in this uncertain environment, manufacturing companies need to possess agility to dynamically and effectively adapt to the changing environment. Agility at the machine level can be thought of as the ability to reconfigure manufacturing machines in response to changing needs and opportunities. This thesis is concerned with a design method for machine level agility for reconfigurable manufacturing machines. This thesis is divided into two portions: a design approach for reconfigurable manufacturing machines and the embodiment of this approach in a computational synthesis example.

In developing this design method, various approaches and reconfigurable systems are presented to develop an overview of the applications and current related research to reconfigurable manufacturing machines. From this related research, a research gap is identified pertaining to the identification of the evolving architecture of reconfigurable manufacturing machines.

The key contribution is the design approach based on co-evolution. This design approach involves the implementation of agent based co-evolutionary algorithms. In this implementation, each agent synthesizes the configuration of a machine for a product in the range of products it is to manufacture and co-evolves with other agents which are synthesizing machines for other products to reduce the reconfiguration cost.

Finally, an in-depth case study of the design approach is presented in which the approach is tested relative to various product changes; thus, showing the advantages of employing an evolving reconfigurable machine architecture. These product changes include batch size variations, geometry changes, and material changes. Hence, the core objective is to identify the necessary reconfigurable manufacturing machine architecture for the range of configurations required for machining various products.

Chapter 1 Introduction

1.1 Motivation

Manufacturing companies today face increasingly uncertain and volatile market demands. Product designs and the required quantities change rapidly to meet the needs of customers. To maintain competitiveness in this uncertain environment, manufacturing companies need to possess agility to dynamically and effectively adapt to the changing environment [9]. Agility is necessary at various levels from the business processes to the manufacturing systems. Figure 1-1 [8] depicts these various levels of agility in an extended enterprise. The necessity for agility has also been articulated as a grand challenge for manufacturing in the world of 2020 by the National Research Council as: “reconfigure manufacturing enterprises rapidly in response to changing needs and opportunities.”

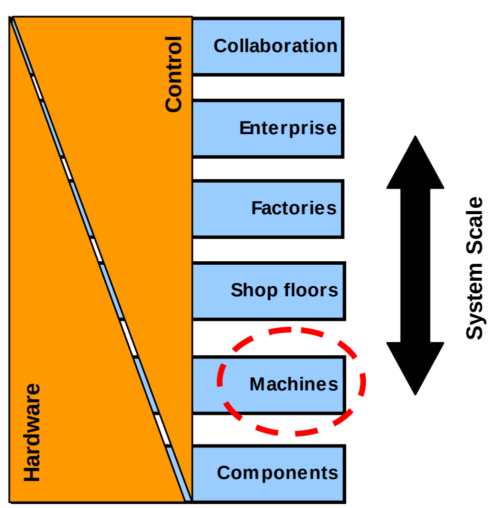


Figure 1-1 – Levels of Agility (adapted from [8]).

This thesis addresses agility at the machine level as shown by the red circle in Figure 1-1. Agility at the machine level can be thought of as the ability to reconfigure manufacturing machines in response to changing needs and opportunities. Accordingly, the research question addressed in this thesis is ‘what should the architecture of manufacturing machines be such that they can be reconfigured and adapted to changing needs and opportunities?’ A design method for the architecture of reconfigurable manufacturing machines is proposed. In this chapter, different types of manufacturing systems are introduced in Section 1.2. Reconfigurable manufacturing systems are defined as the class of manufacturing systems that are relevant to this thesis. Section 1.3 provides an overview of the reconfigurable manufacturing system design problem and Section 1.4 presents the organization of the thesis.

1.2 Types of Manufacturing Systems

Manufacturing systems may be divided into three general categories, dedicated manufacturing lines, flexible manufacturing systems and reconfigurable manufacturing systems. Each category is discussed in this section.

1.2.1 Dedicated Manufacturing Lines

The dedicated manufacturing lines or transfer lines represent a sequential, series of machines used to manufacture a single specific product. Transfer lines may be characterized by a few salient features: a) dedicated transfer equipment, b) low cost per part when demand exceeds supply, and c) high production volume [18]. An example of a transfer line is shown in Figure 1-2. In this example, the product flows through a series of machining operations in the direction of the arrow. At each machine, features are cut into the work piece until all processes associated with that line have been completed.

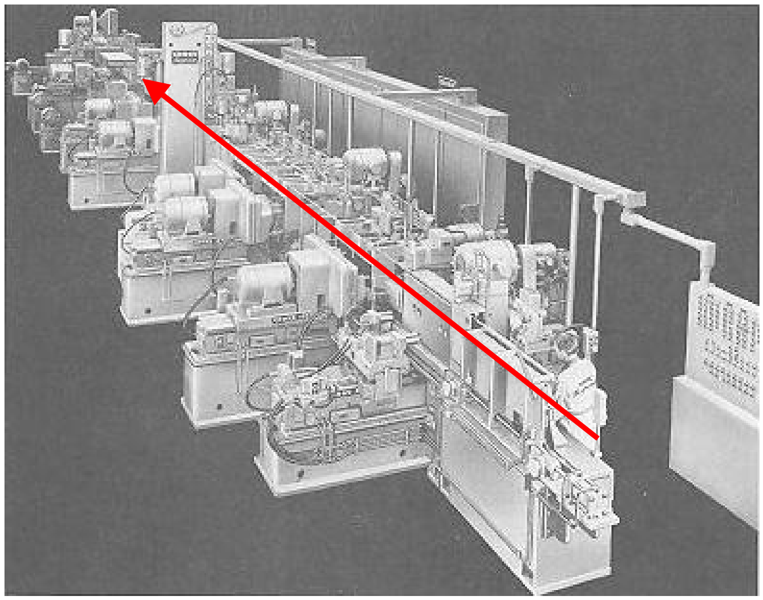


Figure 1-2 – An example transfer line [18].

The transfer line depicted in Figure 1-2 is optimized for a specific product. Hence, in the face of an increasingly dynamic marketplace characterized by globalization and requirements for mass customization, transfer lines fail to deliver the same low cost per part due to excessive supply. To cope with the changing needs of the market, a transfer line could be augmented or supplemented with new machining functionality to achieve new product or volume capability, but the cost is often times too high to purchase, install, and setup new dedicated, specialized machinery to adapt to the changing consumer environment. To answer needs for increasing product mix and fluctuating volume demands, FMSs were introduced.

1.2.2 Flexible Manufacturing Systems

Flexible manufacturing systems (FMS) were created in the 1960’s to achieve a wider variety of production capabilities than traditional dedicated transfer lines. An example FMS is shown in Figure 1-3.

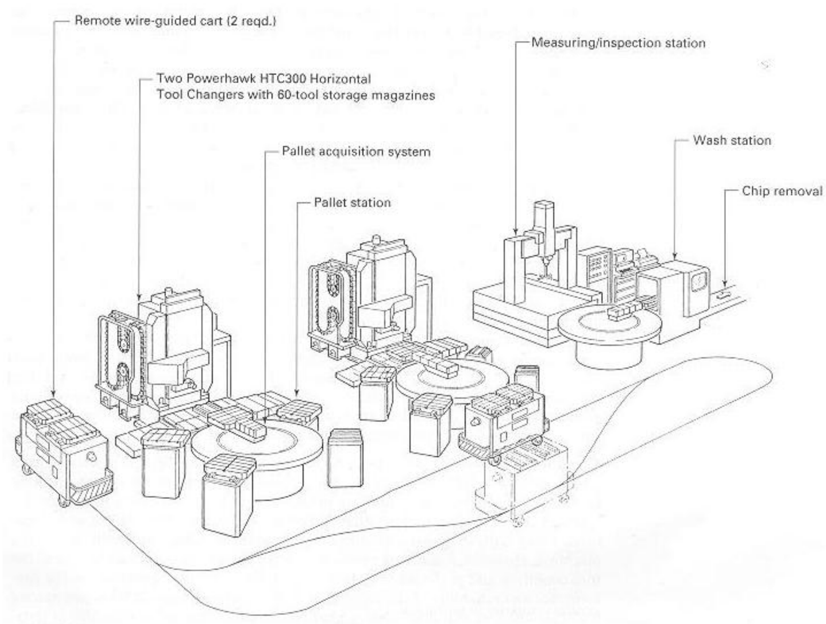


Figure 1-3 – An example diagram of a FMS [18].

FMSs combine the repeatability of transfer lines with the flexibility of computer numerical control (CNC) machines, denoted by the two horizontal milling machines in Figure 1-3. CNC machines represent a class of machine tools which are equipped with computers to store programs, tool offsets, send and receive information, and perform a variety of other data processing functions. In general, FMSs achieve flexibility through the use of this programmable software architecture to quickly change work orders and process sequences. Work orders and process sequences represent the specification of the number of parts to be produced and the series of operations necessary to arrive at the desired part geometry [34]. With these types of features, FMSs typically contain 2 to 10

CNC machines and produce volumes within the range of 3000 to 10,000 parts per year with a 2 to 20 product variations by adjusting machinery or transfer equipment to allow for new process pathways throughout the system [18]. Due to this low throughput and the high cost of general purpose CNC machines, the FMS fails to deliver an acceptable cost per part resulting in a small range of applications in which the advantages of FMS are profitable. To satisfy the shortcomings of FMSs, reconfigurable manufacturing systems were proposed to deal with the volatile, uncertain market conditions and cost per part requirements of consumers.

1.2.3 Reconfigurable Manufacturing Systems

Koren et al. [30] defined a reconfigurable manufacturing system (RMS) to be one that is "designed at the outset for rapid change in structure… in order to quickly adjust production capacity and functionality." This implies that RMSs should not only possess the necessary flexibility to manufacture a large variety of parts, but also be able to achieve high throughput. The authors distinguish between reconfigurable manufacturing systems and flexible manufacturing systems, stating that FMSs are based on general purpose CNC machines. The general purpose nature of the CNC machines in FMSs is attributed to the fact that it is not designed around parts to be manufactured. The general purpose flexibility of FMSs thus results in their high costs. RMSs address this problem by being designed around a group of parts. Therefore, an RMS may be optimized for cost effectiveness for a certain variety of parts. A conceptual graph of RMS cost effectiveness relative to production capacity uncertainty is displayed in Figure 1-4.

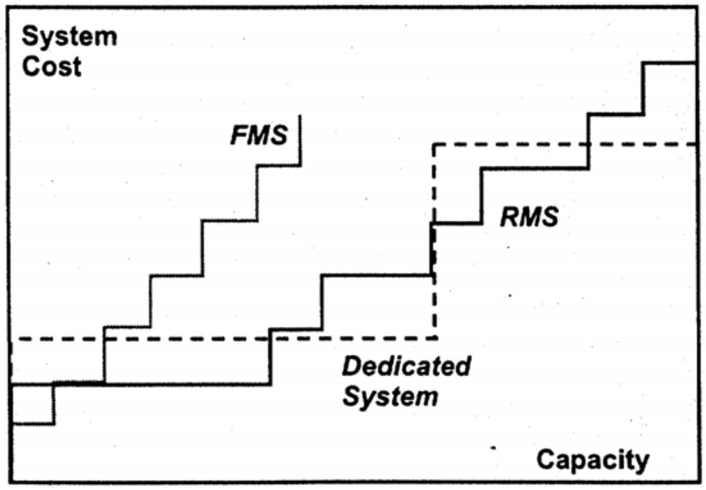


Figure 1-4 – System cost comparison between RMS, FMS, and dedicated lines [30].

For a dedicated system, shown as a dashed line, cost appears to be constant due to the fixed nature of the system relative to the demanded production capacity. When a new capacity is demanded, the overall system cost radically jumps to the required cost of adding a new line to the production system. As for FMS, the system cost increases linearly due to the addition of systems to meet production needs. Furthermore, the FMS cost increases rapidly because new functionality requires the expansion of system by adding entire CNC machines or flexible transfer systems. As for RMS, the inherent adjustable architecture, flexibility for product families and scalability allow RMS to maintain a low relative system cost comparable to that of dedicated systems. By adding minor adjustments to machinery, focusing on accessible product designs within the ranges of system reconfigurability, and scaling up production by the addition of new spindles, axes, tables etc. to an individual system, the RMS avoids costly increases in system capacity by instantiating smaller changes to systems relative to the rather larger changes in a FMS by adding entirely new machines [30].

A comparison of dedicated, flexible, and reconfigurable manufacturing systems is summarized in Table 1-1.

Table 1-1 – RMS comparison to dedicated and flexible systems [30].

	Dedicated	**RMS/RMT**	**FMS/CNC**
Machine Structure	Fixed	Adjustable	Fixed
System Focus	Part	Part group	Machine
Scalability	No	Yes	Yes
Flexibility	No	Customized	General
Simult. Oper.Tool	Yes	Yes	No

The primary concern is developing agile manufacturing systems which incorporate reconfiguration to adapt to uncertain product demands. From FMSs, the focus is creating flexible machining systems based on reusable software and CNC machines for a diverse array of functionality in low volume product demand. This approach is quite different from the RMS perspective which largely relies upon the design and embodiment of individual systems which can facilitate a reconfigurable structure.

1.3 Reconfigurable Manufacturing Machines Design Problem

The reconfigurable manufacturing machines design problem can be divided into two sub-problems, a) the machine architecture design problem and b) the machine configuration design problem. The machine architecture design problem deals with the structure of the machine from which different configurations can be derived. The configuration design problem deals with identifying the appropriate machine configuration for a particular product. This thesis deals with the machine architecture design problem.

As mentioned earlier, RMSs are designed around a range of products to be manufactured. Different ranges of product varieties would therefore imply different appropriate reconfigurable machine architectures. For example, Figure 1-5 shows two ranges of products to be manufactured by two companies at different points of a certain time span. The appropriate machine architecture for company A’s product range is different from the appropriate machine architecture for company B’s product range. The machine architecture in this example is defined by the number and type of machine components. The particular machine configuration synthesized from the components is also shown in the figure.

Further, the range of products which a company manufactures changes and hence, the architecture of the machines should also change accordingly. Figure 1-5 shows the range of products which company A is to manufacture, change from one range to another (Year 1 to Year 2). In this example, two additional spindles and four additional fixtures have been added to the architecture.

From this discussion, it can be summarized that there is a need for a design method that determines the appropriate architecture of reconfigurable manufacturing machines for a particular product range. This method should also account for the change in machine architecture when the range of products to be manufactured changes. Hence, a design method for the architecture of reconfigurable manufacturing machines is presented in this thesis to meet these needs.

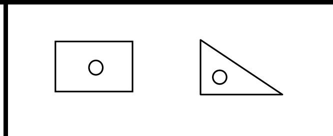
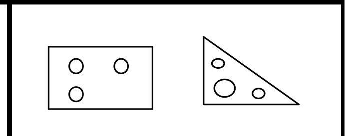
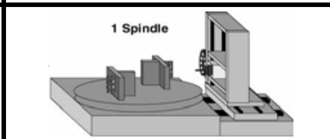
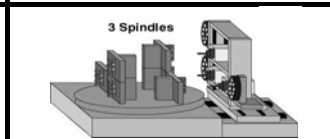
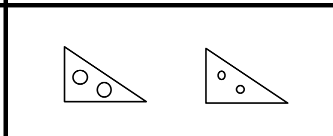
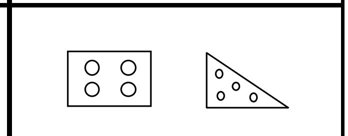
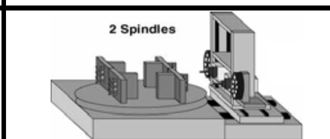
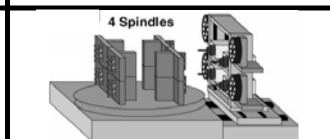
		Year 1	Year 2
Company A	**Product Range**		
	Machine Configuration		
	Machine Architecture	1 Base 1 Column 1 Spindle 1 Table 2 Fixtures	1 Base 1 Column 3 Spindles 1 Table 6 Fixtures
Company B	**Product Range**		
	Machine Configuration		
	Machine Architecture	1 Base 1 Column 2 Spindles 1 Table 4 Fixtures	1 Base 1 Column 4 Spindles 1 Table 8 Fixtures

Figure 1-5: Example of necessity for different architectures (adapted from [47]).

1.4 Organization of Thesis

The rest of this thesis is organized into the following five chapters. These chapters are articulated below:

- **Chapter 2** – Chapter 2 reviews the related research in the design of reconfigurable systems. From the literature review, the objectives of this thesis are defined.
- **Chapter 3** – In Chapter 3, an overview of the design method for the architecture of reconfigurable manufacturing machines proposed in this thesis is presented.

- **Chapter 4 –** In Chapter 4, the proposed design method is applied to the design of reconfigurable milling machines.
- **Chapter 5 –** In Chapter 5, the results of applying the design method to reconfigurable milling machines are presented and discussed.
- **Chapter 6 –** Chapter 6 concludes this thesis with the main contributions made and recommendations for future work.

Chapter 2 Related Research in Reconfigurable Systems

In this chapter, a review of the related research in reconfigurable systems is presented. Three related areas are discussed, a) reconfigurable manufacturing systems, b) reconfigurable robotic systems, and c) design methods for reconfigurable systems. Following the review, the research gaps are identified.

2.1 Reconfigurable Manufacturing Systems

2.1.1 Design Characteristics of RMS

Koren et al. [1, 2] identify six characteristics for the design of a RMS. These characteristics include modularity, integrability, convertibility, diagnosability, customization, and scalability. A definition of each characteristic is as follows:

- **Modularity –** “Design all system components, both hardware and software, to be modular.” [34].
- **Integrability –** “ Design systems and components for both ready integration and future introduction of new technology. ” [30, 34].
- **Convertibility –** “Allow quick changeover between existing products and quick system adaptability for future products.” [30, 34].
- **Diagnosability –** “ Identify quickly the sources of quality and reliability problems that occur in large systems.” [30, 34].
- **Customization –** “Design the system capability and flexibility (hardware and controls) to match the application.” [34]

- **Scalability** – In designing for scalability, the RMS should be able to meet capacity fluctuations to support the changing demand trends. Furthermore, this type of scale-up must occur quickly to ensure that adaptation to the new market demand is met in both a cost efficient and timely manner [26, 47].

2.1.2 Design Methods for RMS

In this section, RMS design methods which incorporate the aforementioned design characteristics are presented and discussed in detail. Three types of design methods for RMS are identified by Bi et al. [8]: a) architecture, configuration, and control design. Architecture design refers to the design of machine components and interactions. Configuration design refers to the identification of single machine configuration or system level, multiple machine configurations to accommodate product requirements. Control design relates to the design of software which governs the operation of the machining system from an entire manufacturing system level down to individual component control. In this section, architecture and configuration design methods for RMS are reviewed.

The foundational work on a design methodology for a RMS is credited to Koren et al. [29]. The design methodology involves the justification of a need for RMS through life cycle assessment. If the RMS is required, a method is posed which first analyzes system level design concerns such as integrability, modularization, and convertibility. Once preliminary system level design decisions are made, the method directs the designer towards a machine-level design. This phase of design introduces design for modularity, integrability, customization, convertibility, and diagnosability. The end result is a RMS. This method is shown in Figure 2-1.

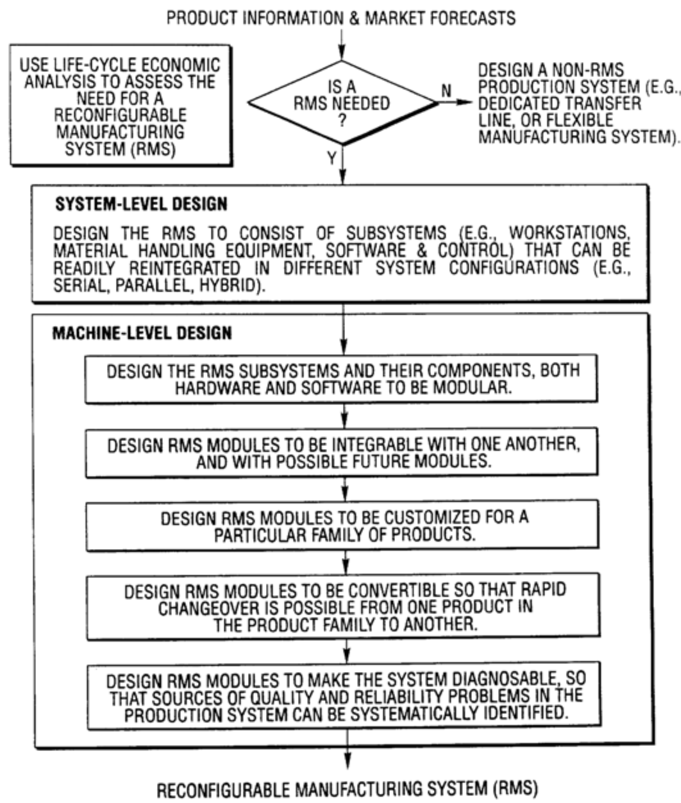
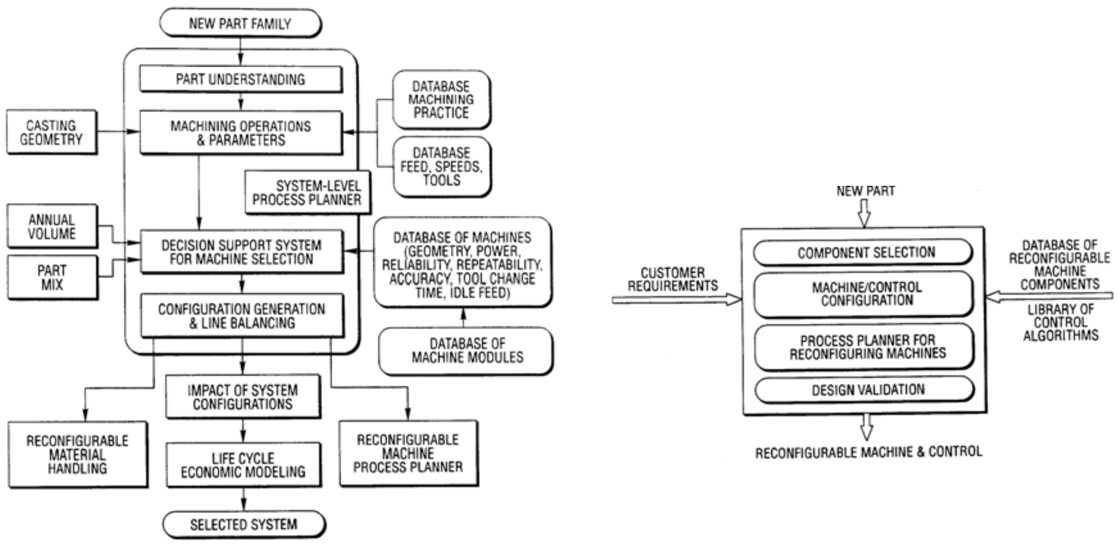


Figure 2-1: General design methodology for RMS [29].

In this design methodology, the author identifies five of the core characteristics of RMS, but gives no guidance to the designer towards accommodating the design characteristics in the method. This method is further embodied into system level and machine level design methods. System level design is concerned with the satisfaction of product family demands. System level design identifies the configuration which will best fit the purpose of the demanded product. Once a system level configuration is identified, it may be necessary to search for convertibility at a lower system scale to identify a means to gain access to higher levels of convertibility. Machine level design involves the identification of acceptable machine modules, configurations of modules, and process planning for that machine configuration. Once these machine level decisions are made, a machine can then be converted to accommodate both the machine and system level configuration requirements. These design methods are shown in [29].



a) b)

Figure 2-2: a) System Level Design b) Machine Level Design [29].

The system level design methodology provides a detailed process of design decisions which must be made to arrive at an appropriate machine configuration. The machine level design methodology transfers the level of convertibility to a lower scale. The machine level design method has been extended by the work of Spicer and co-authors [47].

The design methodology for RMS has been augmented to include scalability in the consideration of the primary design characteristics for RMS. With respect to scalability, it is stated by Spicer and co-authors [47] that “None of the machines is truly scalable.” This fundamental concept of bounded scalability is due to limits on monetary considerations and shortcomings of current design methodologies. The author presents a design method for the scalability of a RMS both at a system and machine level. These design methods were used to develop the designs displayed in Figure 2-7 of Section 2.1.1. In this design methodology, the author presents functional modules such as spindles, fixtures, and tool changers at a machine level and workstations, transfer slide ways, and fixtures at the system level. The author bounds scalability by placing constraints on the module level, type, and positions. These constraints are discussed in the context of the designer's solution representation displayed in Figure 2-3.

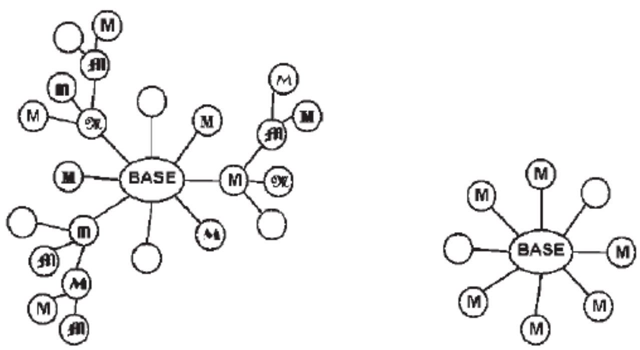


Figure 2-3: Scalable design representation [47].

The level of module is the distance away from the base. The first level is directly connected to the base, the second level is connected to the first and so on [47]. The module type is determined by the designer as the available module designs which comprise the solution space. The position of the module is shown by the blank modules in Figure 2-3. These represent open module interfaces for scaling machine capability. Aside from scalability, the authors further expand upon the idea of a scalable, multi-spindle machine tool by arguing that multi-spindle machines increase throughput and reduce conversion time, while reducing reliability due to increased complexity. Other work in the area of configuration identification relates to the mapping machine configurations to operations and evaluating a system based upon the capital cost of the configuration at the current time.

In another design method, Youssef and ElMaraghy [55] organize a machining system into machines, machining clusters, and operational cluster setups. In this instantiation, machining clusters represent sets of machines which are combined into operation clusters. These items group into a configuration which is a series of stages comprised of operational clusters. To account for convertibility, the authors introduce a new metric referred to as reconfiguration smoothness [55]. The smoothness metric is used to evaluate configuration closeness based upon cost, time, and the effort required to convert one configuration of operating clusters to another configuration of operating clusters. The overall equation involves evaluating market level, system level, and machine level reconfiguration smoothness as denoted by [55] –

$$RS = \alpha TRS + \beta SRS + \gamma MRS$$

where α, β, and γ represent weights for the relative amount of cost, time, and effort for reconfiguration at the varying levels of a manufacturing enterprise. MRS, SRS, and TRS represent the machine level, system level, and market level reconfiguration smoothness. In the MRS equation, there are two variables MRS_d and MRS_o which represent the changes to machine modules and changes to operation clusters, respectively. This equation is given as [55]–

$$MRS = \nu MRS_d + (1-\nu) MRS_o$$

where ν is a weight between [0 1] that evaluates the reconfiguration effort to change configurations. For a detailed explanation as to how MRS_d and MRS_o are calculated refer to [55]. The SRS and TRS are calculated in a very similar fashion. The primary difference is the weighting values and focus of machine addition and removal for TRS and handling systems, machines, and stages (groups of operation clusters). For these equations refer also to [55]. The authors use a genetic algorithm (GA) [23] to identify the feasible, enterprise configuration which yields a minimal capital cost for the configuration at the current time for a given workpiece. The author places constraints on the algorithm in terms of configuration length and width of the series of stages. The optimal configuration results in a configuration period which represents the time by which a current configuration is required by given product demand requirements, production capacity, and the total cost of the system.

Further work on quantifying convertibility is addressed in a different fashion by Maier-Speredelezzi and co-authors [11]. Convertibility can be evaluated at the system, configuration, and machine level. For the purposes of this thesis machine level convertibility is discussed. For a detailed explanation of system and configuration convertibility quantification refer to [33].

Machine convertibility is calculated on the premise that machines may contain unique functionality which can lend to the convertibility of a system. The machine convertibility is given by [33] –

$$C_M = \frac{\sum_{i=1}^{N} C_M^{'}}{N}$$

where N is the number of machines and the $C_M^{'}$ represents a ranking metric from 1to 10 which identifies whether a machine is flexible, reconfigurable, or dedicated.

Maier-Speredelezzi [33] further express convertibility and scalability using the Analytic Hierarchy Process (AHP) [42]. The AHP is a method of eliciting and evaluating decision making. The authors use an adapted AHP to evaluate system performance based upon productivity, quality, convertibility, and scalability relative to some input workpiece and a certain batch size. From this analysis the author shows that a purely parallel configuration represents the best configuration relative to the quality, productivity, convertibility, and scalability metrics chosen. These findings conflict with the results found in [26]. These conflicts are due to the different metrics for evaluation and chosen solution approaches. Similar to Maier-Speredelezzi, Moon and Kota present a method for convertibility, although in this method the focus is on a more embodiment focused approach.

Moon and Kota [35] implement function structures in a unique way such that the machine structure is generated from the required process plan. In this implementation, the process plan is determined by screw theory to determine the tool paths. The function structure is now generated based upon the required operations such as turning or drilling. Based upon this functional hierarchy, the tooling modules are selected. The advantage of implementing function structures in this way is to enable the identification of various means to accommodate that function through specific working principles. Furthermore, the motion path of the column, spindle, and tooling module may be tied to that function structure such that the process plan is automatically created. Moon and Kota's work represents a practical means of identifying machine configuration from a tool path.

From this discussion, a review of the current RMS configuration design methods has been introduced. These design methods include general approaches for RMS, convertibility based approaches, and functional design synthesis algorithms. For a more comprehensive survey of current design methods refer to [8].

2.1.3 Designed Hardware

Based on the proposed design methods, a number of hardware systems have been designed and embodied. In this section, reconfigurable machine tool designs and concepts are introduced which incorporate the design principles articulated in Section 2.1.1, specifically convertibility.

Koren [28] presents a design for a reconfigurable machine tool based upon the modularity of spindle and cutting tools. The spindle and cutting tools are placed in a T-slot which enables the cutting tools to be reconfigured into positions conducive for the process plan in operation. A diagram of this machine is displayed in Figure 2-4.

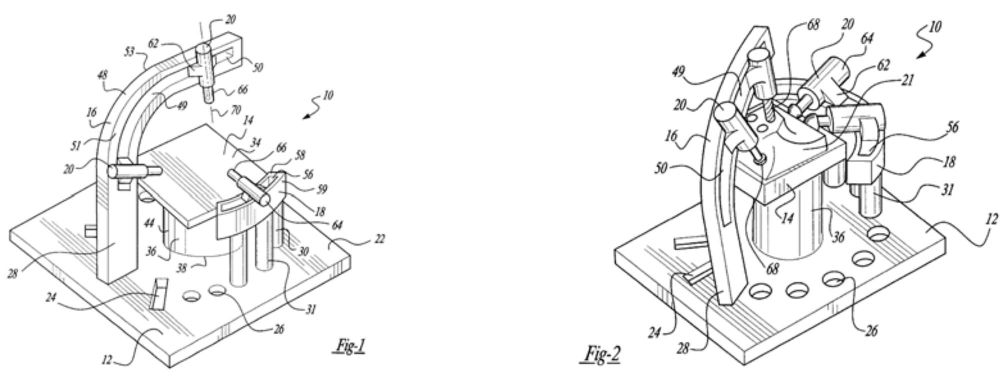


Figure 2-4: Diagram for a reconfigurable machine tool [28].

Two potential machine tool configurations can be seen in Figure 2-4. As in a common milling or drilling operation, the table provides mobility to the workpiece in the x, y, and z directions. The uniqueness of this system is derived from the inclusion of slots and holes for modularity. Slots and holes for columns are available for the location of vertical legs used to guide the motion of tooling.

Moon et al. [35] state "*machine tools are created to fit the function and the performance required to perform a set of operations*." This statement is counter to the typical design of machine tools; whereby, a machine tool is designed and a process plan is developed after the machine is designed. From this general idea, the structure of a machine is generated using a mathematical formulation based upon screw-theory to develop a process plan which can then be used to generate function structure (operations to be performed) and related structural graph to meet that specified process plan. An example of a variety of machine configurations derived from a single structural graph is shown in Figure 2-5.

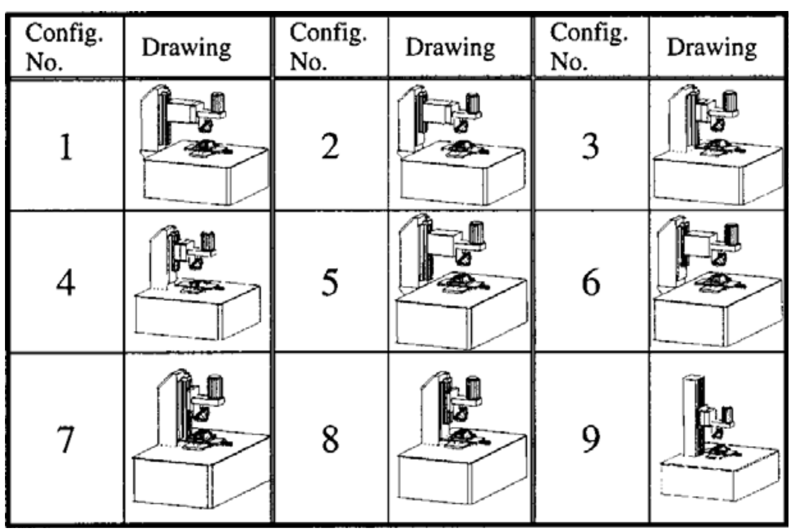


Figure 2-5: Configurations from a single structural graph [35].

In this figure, a number of different fixture and column configurations are shown. The primary differences in these architectures are the different interfaces used to fasten the product or column to the base structure. Through these different interfaces, various modules could be attached or detached to enhance the changeover capabilities of the system architecture. To ensure the creation of coherent, modular machines a connectivity matrix is used in conjunction with a module library to sample the appropriate components [35].

A system level design for diagnosability is accounted for in Koren et al. [29]. In this design, a series of sensors are placed such that sensor movement is capable about the part such that the part can be automatically inspected. The novelty of the design lies in the slots for sensors. The slots allow for adjustment to sensing equipment such that new equipment can be added or old equipment be relocated. A diagram of this design is displayed in Figure 2-6 [27].

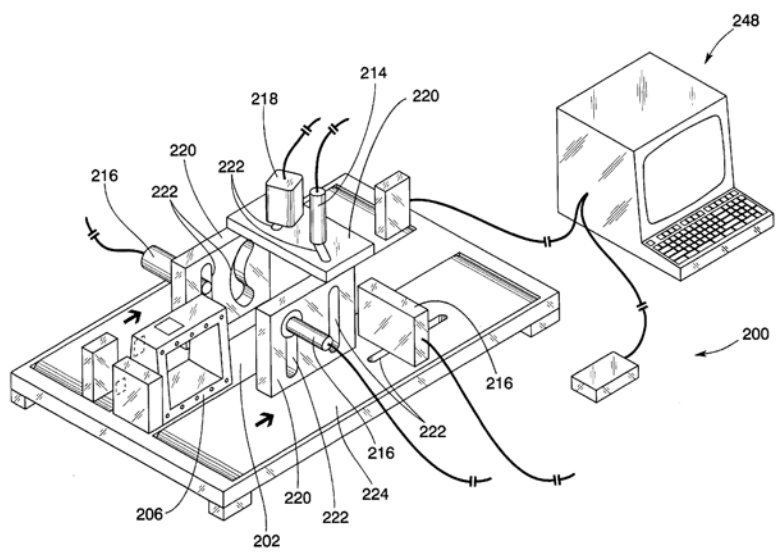
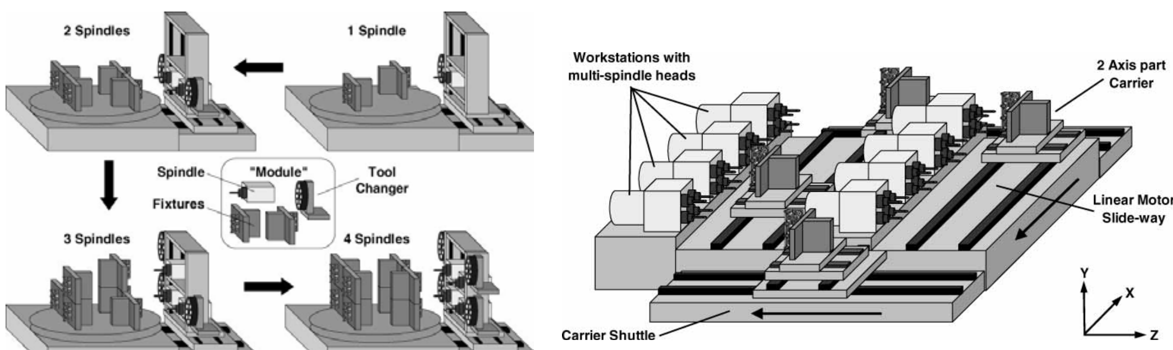


Figure 2-6: A design for a reconfigurable inspection system [27].

The part is denoted as 206 and rides along a conveyor system in the direction specified by the arrow. The sensors are located at 216 and 218. As can be seen, the sensors are mounted in slots which allow for reconfiguration when a workpiece design change occurs. The process data for this system is then stored in a computing system for further decisions about the current manufacturing process.

Further work was carried out by Spicer et al. [47] to present a numerical example of production rate versus the number of modules contained within a reconfigurable machining system. In this instance, the author uses four metrics: a) capacity increment size, b) lead-time, c) cost per unit of capacity, and d) floor space per unit of capacity to investigate the conceptual design of two machines. These machines are displayed in Figure 2-7.



b)

a)

Figure 2-7: Conceptual machine tool designs [47].

In these designs, scalability is achieved in two ways: a) the modularity of system components and b) the positioning of the transfer system. In Figure 2-7 a), the system is designed to scale-up the number of fixtures, spindles, and tool changers; thus, increasing production capability. The author couples this set of components into a functional module. The author extends the idea of scalability to a system with multiple workstations equipped with multiple-spindles. The transfer system carries the fixtures and components through the system introducing the workpieces to a series of possible operations at each workstation [47].

In the work of Heisel and Meitzner [22], the authors introduce a machining system which embodies all six design characteristics, into a drilling and milling machine by using a machining cell architecture. A single manufacturing cell design is displayed in Figure 2-8.

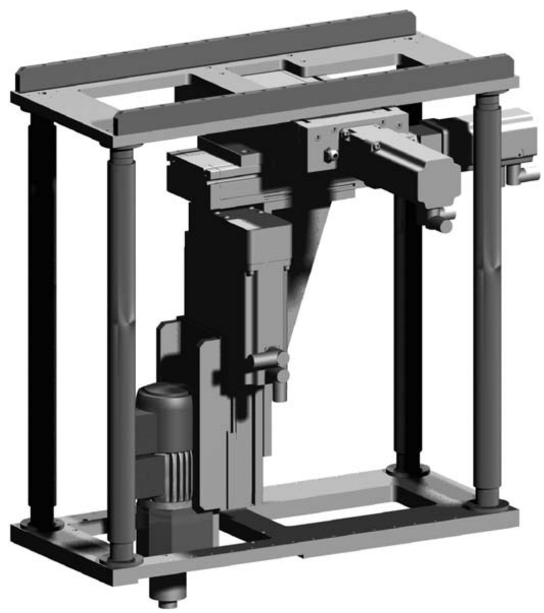


Figure 2-8: Drilling and milling machining cell [22].

Modularity is integrated into this machining structure by means of the addition of the machining frame about which the main spindle and axes are supported. This machining frame is designed to be assembled and disassembled using a minimal amount of tooling for ease of convertibility. Furthermore, the frame may be attached to other cells such that floor space is conserved and scalability is enabled. Customization is also enabled by the cellular architecture in that the cells can be adjusted in terms of size to accommodate various product geometries and features. Integrability is addressed by the author by the ability to change out the main spindle for other types of units. This type of integrability is limited to machining systems which operate in a similar working principle to milling and drilling operations. Diagnosability is tied to peripherals which the author specifies to accommodate the software associated with controlling the machining system. Overall, this machining concept addresses reconfigurable machining from a more traditional perspective of a machining cell. A similar cellular proposal is presented by Morey [17] in the following review of serial manipulators in machining operations.

Morey [36] claims that “Articulated arm robots could be an alternative to CNC-style machining centers in some applications.” Typically, serial manipulators are popular in applications such as deburring, welding, and assembly applications. Another application of robot arms is that of a Stewart platform which can hold tolerance up to a ± 0.025 mm accuracy. This configuration acts much like a traditional vertical milling arrangement. The author goes on to mention that serial manipulators are already present in some plastic, wood, and water jet machining tasks in which strict tolerances are not present. By implementing these types of systems, modularity, integrability, and convertibility is granted through work cells and end effectors with connection interfaces for the acceptance of new hardware. The articulation granted by the many axes of robot arms provides the system with customization capability in the context of changing the machine program to produce new workpiece variants. Finally, scalability can be achieved by the addition of new works cells. A couple of robotic machining examples are shown in Figure 2-9.

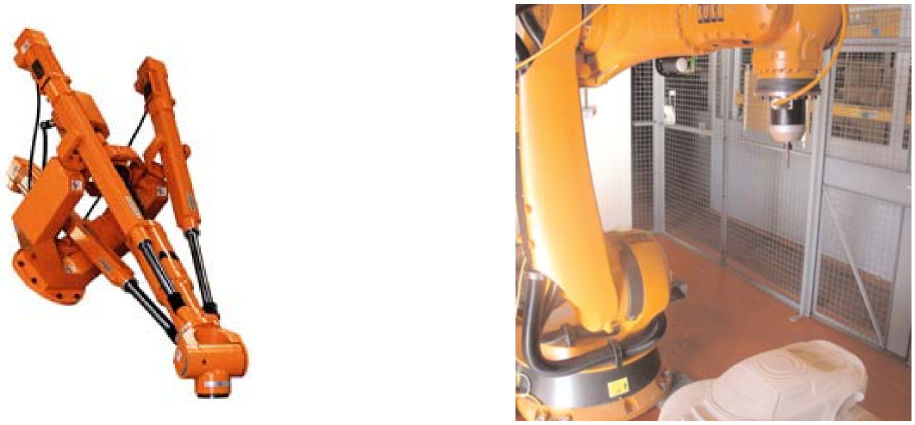


Figure 2-9: a) ABB machining prototype[2] b) Robotic CNC Solutions Kuka prototyping machine [3].

Robot manipulators provide great promise in the context of enabling greater satisfaction of the design characteristics of RMS design requirements. Further research in the context of adaptive control is required to grant diagnosability to accommodate quality and reliability issues associated with machining operations.

Landers et al. [32] present two designs for machine tools which enable the accommodation of product changes. In Figure 2-10, a prototype reconfigurable machine tool (RMT) is displayed. In this design, the fixture assembly represents the primary source of convertibility in the system architecture. The tooling is placed in a configuration with 3 degrees of freedom (DOF) which allows the spindle and bit to be indexed in such a way that allows the product to be manufactured.

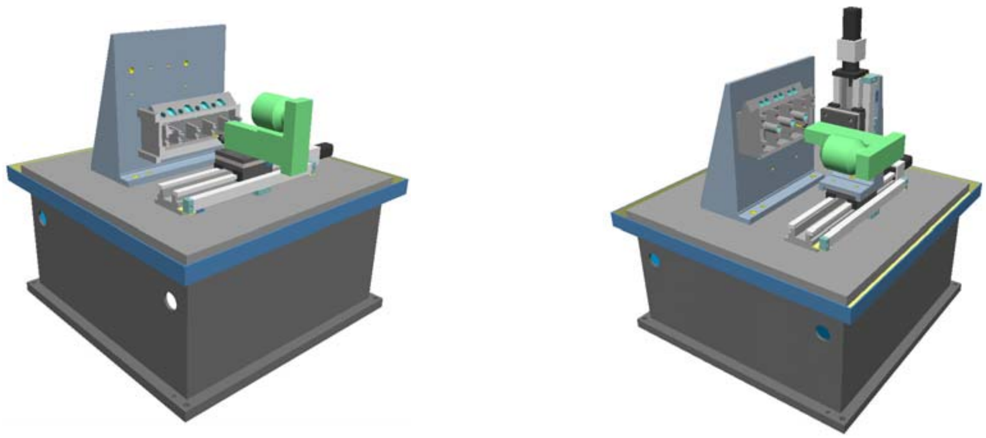


Figure 2-10: RMT prototype [32].

This relatively simple prototype could not be easily scaled without adding new machines. This characteristic is detrimental to the machines utility in the context of a RMS.

Landers et al. [32] present another set of designs for RMSs. These designs are pictured in Figure 2-11. In the first design shown on the left, the system is largely focused upon workpiece geometries which feature angled faces. This type of machining system achieves this by placing an arched pathway for the tooling to follow while being indexed. For this type of geometry, this system would work well in the context of convertibility and customization. As for scalability, it appears that the design would require additional machine lines to truly scale-up production capacity. While this is a common feature in the manufacturing literature, this may not present the best accommodation of the design requirements. Another design is presented on the right of Figure 2-11.

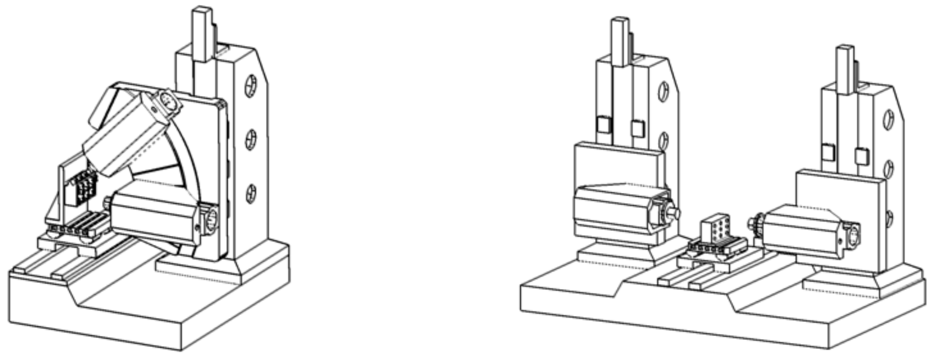


Figure 2-11: RMT design [32].

This design represents a machining system which is conducive for the concurrent machining of features in a single component. In this type of cutting, the machining dynamics become very complicated. Hence, the complexity of the machine must increase. To deal with this complexity increase, modularity is embodied by using standardized tooling modules.

For a more in depth overview, refer to [8] and [30]. Similar to the work in RMS, a great deal of work has been performed pertaining to structural configuration design in the robotics literature, which will be reviewed in the following section.

2.2 Reconfigurable Robotics

Self-reconfigurable robotics is a field of robotics in which robotic systems are designed with a modular architecture capable of autonomously reconfiguring their structure into different configurations to achieve enhanced functionality. Within this field, a great deal of work has been performed to address the appropriate system configuration for a specific means of locomotion. Several of these systems are reviewed to analyze the relationship between the robotic structure and means of reconfiguration.

These reconfigurable systems include Molecubes [38], PolyBot [49], Superbot [43], and M-TRAN [37].

Mytilianaios et al. [38] present a self-replicated, reconfigurable robotic system known as Molecubes. The hardware design of this system includes a modular architecture comprised of a series of identical cubes. Each cube contains one DOF for rotation about its long diagonal plane. Furthermore, each cube contains six electromagnets which act as standardized interfaces for the convertibility of the structure into other configurations. A diagram of this system is shown in Figure 2-12 [56]. In this diagram a), a molecube and model of the interior components of a molecube are shown. Also displayed in b) is the self-replication process that this system goes through to create another set of molecubes. To complete this replication, a series of configurations must be accomplished to retrieve raw materials (molecubes) and place materials.

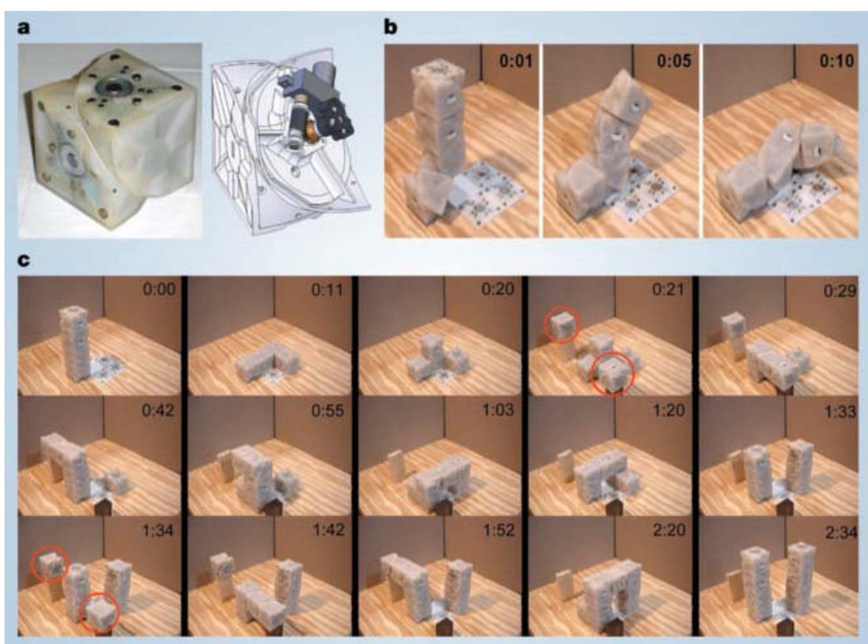


Figure 2-12: Molecubes self-replicating process [56].

The configuration identification process is driven by an evolutionary algorithm which searches various configurations for an acceptable replication. This process is eased by the modular architecture which is very similar to several other reconfigurable robotic systems. One of which is Superbot a reconfigurable robot with multiple modes of locomotion.

Salemi and coauthors [43] developed a modular, reconfigurable robotic system coined Superbot. As stated by the authors, the design philosophy of Superbot is as follows "…develop flexible, powerful and sturdy modules that can efficiently perform tasks in an uncontrolled environment…" The Superbot is comprised of identical modules much like the Molecubes, except in this case each Superbot is more capable of various modes of locomotion. This functionality is born of the torque available in each module to lift several neighboring modules, sensing capabilities to locate threats and objectives, the ability to self-locate a charging station, and the presence of distributed control software. For a more in depth explanation of the hardware in terms of components such as controllers, gearboxes, and DC motors refer to [43].

As for the configuration design methodology or locomotion identification, the control gaits were created by two methods: human designed or designed by a genetic algorithm. In both cases the designers use select configuration (shape of the modules), slope (slope of the terrain), obstacle (type of terrain), speed (speed of the mode), turn (mode turning capability), energy (efficiency of power consumption), and recover (fault tolerance and recovery) [44]. Overall, the modules combine their behavior in a distributed and parallel fashion to coordinate collective behavior towards reconfiguration in response to an environmental stimulus [43]. A similar system to Superbot is M-TRAN which uses an evolutionary algorithm for reconfiguration.

The hardware design for M-TRAN includes a modular structure in which each module supports reconfiguration. As can be seen in the left of Figure 2-13, each module contains two DOF which allow for rotation about a central link. An example of this hardware is shown in Figure 2-13. Using this hardware architecture, the method of reconfiguration is simplified due to the magnetic interfaces and simple square faces [37].

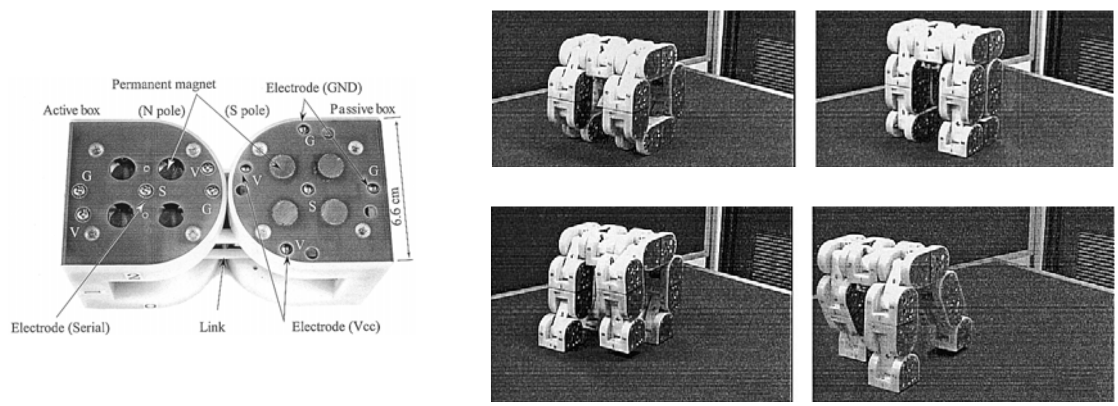


Figure 2-13: An individual M-TRAN module and a configuration of modules [37].

In the right of Figure 2-13, a series of the robot configurations is shown. This reconfiguration is enabled through the generation of motion sequences by a GA. To implement the GA, solutions are evaluated using metrics such as walking speed and efficiency [54]. By using these metrics, a locomotion gait is synthesized which is both functional and effective. For an overview of these experiments refer to [31, 54]. A similar system which uses gait control tables for reconfiguration is Polybot.

Polybot is dubbed as a ‘n’ modular robotic system with the capability to assume a variety of system architectures [52]. These system architectures are used to satisfy the demands of a changing environment. This versatility is granted through the presence of a large number of identical modules which assume various roles during a series of configurations.

To design these modules, one design principle is used: simplicity [19]. Design simplicity is achieved by creating one DOF modules which are relatively useless by themselves, but can achieve far greater functionality in groups. In the design of this hardware, two connection plates are positioned for docking purposes. These interfaces contain pins and chamfered holes for the guidance of safe docking, as shown in Figure 2-14.

Using an array of modules, an interesting behavior can be observed in terms of several walking gaits. To control the motions, pre-calculated gait control tables are created for specific system configurations [19]. For more information on the connection or disconnection of modules or the individual gait motions refer to [50, 53].

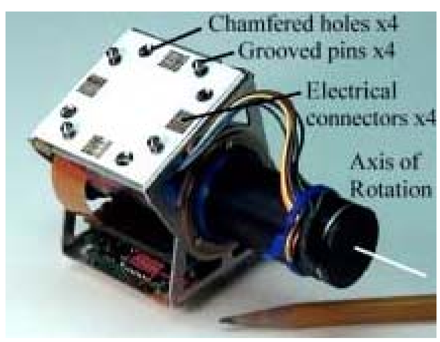


Figure 2-14: Polybot module with labels [19].

For reconfigurable robotic systems, several hardware designs have been presented with reconfiguration capability. In these systems, structure is specified by the designer and configuration is typically found by searching the gait control design space using evolutionary algorithms. For further information on these types of systems refer to [51].

2.3 Design Methods for Reconfigurable Systems

In this section, several approaches to the design of reconfigurable systems are reviewed. Many of these methods are based upon multi-objective optimization.

De Weck et al.[17] present a modular space system concept based on the ideas of reconfiguration and reuse. The modularity for this conceptual system structure is based upon an octahedron. In this study, the reconfigurability of the system is the number of unique configurations which can be created from the selected modules. For this conceptual design method, there are four phases: a) definition of a "Point Design", b) subdivision of modules, c) design space exploration, and d) a feasibility check. A diagram of this method is displayed in Figure 2-15.

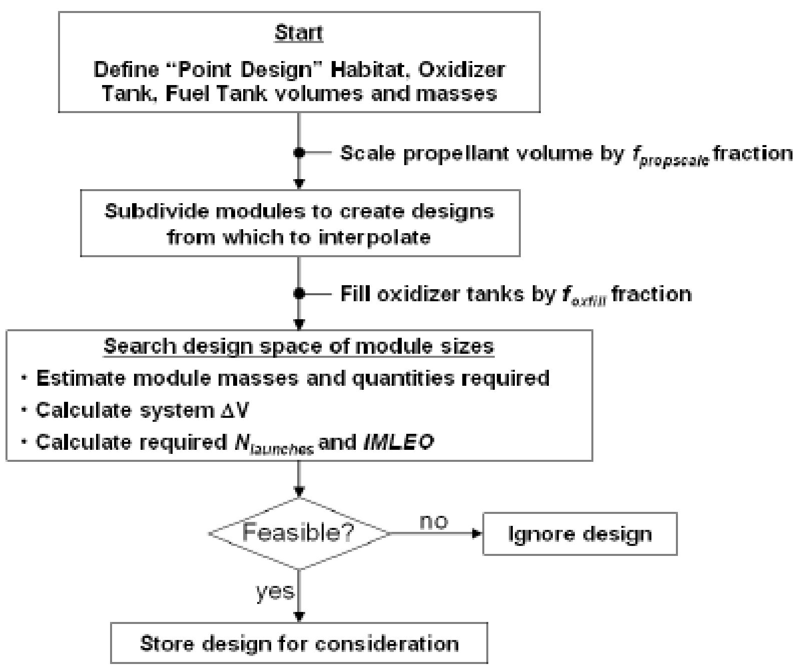


Figure 2-15: Conceptual design method for the structure of a point design [17].

The design method presented in this case study is tailored to a Transer and Surface Habitat (TSH) for a NASA Mars mission [3]. To begin, a point design for a system of interest is selected. This point design sets the high-level parameters such as mass and volume in this case. From this point design, an objective function, design variables, and design constraints are selected for the desired concept exploration. Once these decisions are made, the volume of the structure is interpolated such that a volume and mass can be estimated for the selected modules. When the modularization is determined for the given structure, the design space of solutions is searched such that an optimal configuration is synthesized based upon the objective function selected. The concept is then checked for feasibility. If the concept is determined to be feasible, the concept is stored for future consideration [17]. An example of a modular, synthesized system is displayed in Figure 2-16.

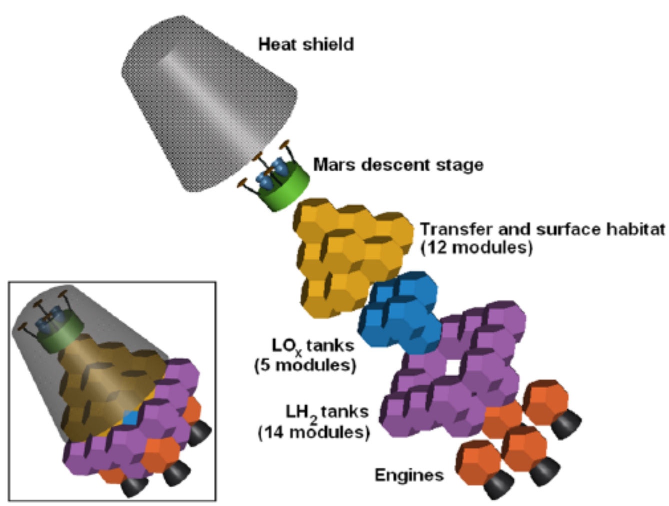


Figure 2-16: An example of modularization using the selected approach [17].

In other work by Siddiqi et al.[46], a concept for reconfigurable systems as a means to achieve adaptation is presented. In this implementation, the authors propose a Non-Homogeneous Markov Model to accommodate reconfigurability in a system architecture. By selecting this approach the authors assume that the process of reconfiguration involves configuration state transitions based upon probabilistic performance metrics which define at discrete time intervals what configuration should be realized .[46] In the case study used to demonstrate this approach to reconfiguration, the author selects a wheel for a roving vehicle in which the surface characteristics of the terrain are unknown a priori. In this case study a simulation of the ground interaction with the wheel is created using a conceptual design of a reconfigurable wheel with the capability to expand its outer diameter. With this system, the diameter of the wheel expands or contracts to fit the current soil conditions [46]. A further extension of this work includes the identification of the working space of spare components for reconfigurability and commonality.

In this approach to reconfigurable systems design, Siddiqi and de Weck [45] take a different approach to system level design which includes a set of different machines with reconfigurable or dedicated repair components. With a system failure, the reconfigurable or dedicated repair components are introduced to the system to fix the system problem. In the case study, the systems include a surface habitat, ascent/descent vehicle, and an all terrain vehicle. For the purposes of this study a single arbitrary dedicated spare and reconfigurable spare repository are used. Each type of spare is assigned a failure probability. A discrete event simulation is used to identify the availability of components or the amount of components which are available to use for repairing a failed system. Furthermore, the simulation allows for trades between non-operating and non-functional systems. In summary, the simulation relies upon the principle of necessity for repairing or replacing systems. The simulation is run for a certain length of time while accounting for the failures of systems and application of repair components from repositories. From the simulation, it is shown that availability is significantly higher than the dedicated repair components for increasing failures. This shows that the reconfigurable repair components are advantageous due to their cross functionality between systems. The authors further elaborate that the increase of system reliability may not be feasible in light of the cost of more complex reconfigurable components [46]. In another example, Nadir et al. [39] present an optimization tool for synthesizing the topology of a truss structure for reconfiguration.

In this design approach to reconfiguration, a truss structure is optimized for reconfigurability and uncertain loading conditions based upon a working space of simple truss elements. To argue the need for reconfigurable structures, the authors present three cases studies which involve a structural optimization for each loading case, structural optimization for robustness to both loading cases, and a structural design which accounts for reconfiguration. A diagram for the loading cases is displayed in Figure 2-17.

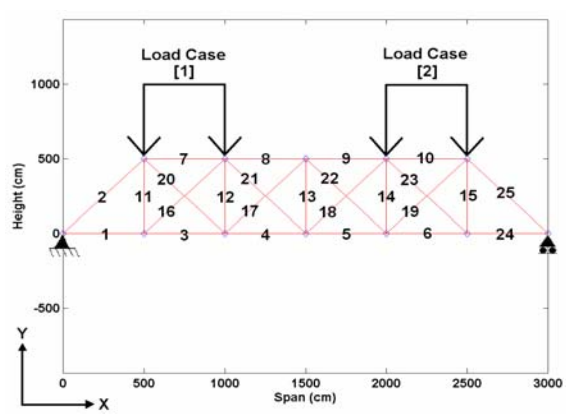


Figure 2-17: Loading cases with arbitrary truss structure layout [39].

For the optimization, a gradient based optimizer is used for structural optimization in conjunction with a random search of configurations. The objective of the optimizer is to minimize the cross sectional area of the structure thus minimizing the required material and manufacturing cost of the structure. To enable reconfiguration, another random search is used to reconfigure the components from an optimal configuration to another configuration which satisfies structural constraints. If the structural constraints are not met, the algorithm reiterates the random sampling process. A process diagram of the reconfiguration is shown in Figure 2-18.

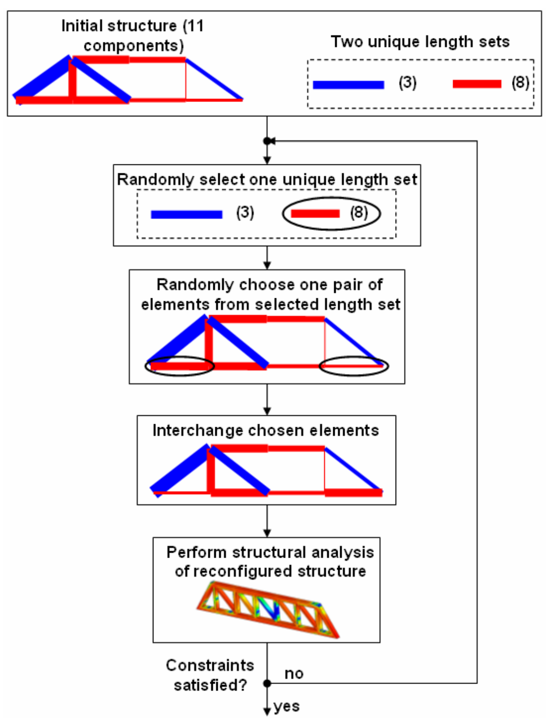


Figure 2-18: Reconfiguration process for truss structure [39].

From this study, the authors show that manufacturing cost is greatest for the robust solutions and smallest for the custom designed truss structure. The manufacturing cost for the reconfigurable structure is moderate. In instances where both loads are required, the two separate custom structures are far more expensive than the robust and reconfigurable solutions which are roughly equal. In this context, the authors show that the reconfigurable structure is a moderately good solution in the event of one loading condition and that it is optimal in the event two loading conditions by a marginal amount.

Another approach to reconfigurability is presented by Olewnik et al. [40]. This approach represents another means of maintaining flexible system performance. This flexible system performance is achieved by identifying an adaptable structure and the associated cost of a flexible design. The general method is based upon an optimization problem which in this case is a minimization of the time required to traverse a race track [40]. The author uses an objective in the context of a race car example which has design variables of a center of gravity, roll stiffness, and down force. In this context, the approach attempts to find the best mix of adaptable variables which account for the various configurations of a race car that would provide performance increases on a racetrack. The authors try three different methods to find a best path of reconfiguration in a design space of variables.

For the first method, the authors map a Pareto frontier by doing a grid search of the design space. Once a Pareto frontier is identified, the authors argue that a certain degree of optimality may be obtained by tracing the Pareto frontier from a maximum at one design variable to a maximum at another design variable. In this instance it may not be the best path of reconfiguration due to uncertainty in the control of the reconfiguration [20].

As their second proposed method, the authors develop a better approach to locating a best reconfiguration path by taking the linear vector between two Pareto extreme points. In taking such a route, the system deviates from the non-dominated front and might violate system constraints; therefore, a solution of this nature would not be physically plausible [20].

The third and final method presents a hyper box based approach whereby the design space is windowed about design points such that the path to reconfiguration is discovered. This involves minimizing the objective function within the hyper box while maintaining a minimum deviation from the Pareto frontier. This process is iteratively performed until the design points reach the Pareto extreme point or the point which satisfies reconfiguration [20]. From these methods, the authors show that the hyper box iterative approach proves to find the path of reconfiguration which provides both objective and deviation minimization.

2.4 Research Gap

The research question addressed in this thesis is 'what should the architecture of a manufacturing machine be such that it can be reconfigured and adapted to changing needs and opportunities?' It was identified that the architecture of the reconfigurable manufacturing machine should be based on the range of products to be manufactured. It was further identified that the architecture of the machine should evolve as the range of products to be manufactured evolves.

The related research review presented a number of methods for determining the architecture of a system such that it can be reconfigured and adapted to changing conditions. These methods were reviewed from the areas of reconfigurable manufacturing systems, reconfigurable robotics and design methods for reconfigurable systems. In the area of reconfigurable manufacturing systems, efforts have identified general characteristics of reconfigurable manufacturing systems and presented general and detailed methods for designing reconfigurable structures. Metrics for measuring convertibility between manufacturing machine configurations have also been proposed. In the area of reconfigurable robotics, various detailed structures of robotic systems for reconfiguration have been proposed and implemented. Much of the efforts in reconfigurable robotics have focused on shape reconfiguration. In the area of design methods for reconfigurable systems, various optimization based methods have been proposed for determining the appropriate architecture for reconfigurable systems. Although all of the proposed methods for determining the architecture of reconfigurable systems in these three areas are viable for determining the architecture of a reconfigurable manufacturing machine, an area that has not been sufficiently addressed is a method for the design of evolving machine architectures. As mentioned earlier, machine architectures in a reconfigurable manufacturing system evolve as the range of products to be manufactured evolve. Therefore, *there is a need for a design method for determining the evolving architecture of reconfigurable manufacturing machines.*

To arrive at a design method for determining the evolving architecture of reconfigurable manufacturing machines, it is hypothesized that biological evolution could inspire and present an approach for designing evolving architectures. A design method for evolving machine architectures based on biological evolution is presented in this thesis.

Chapter 3 A Co-evolutionary Multi-Agent Method for Designing the Architecture of Reconfigurable Manufacturing Machines

In this chapter, an overview of the design method for reconfigurable manufacturing machines inspired by biological evolution is presented. This chapter first discusses biological evolution in Sections 3.1 and 3.2. Section 3.3 presents the concept of an evolving factory. In Section 3.4, the design method is presented. Section 3.5 concludes the chapter.

3.1 Biological Evolution

Evolution is a process in biological systems involving the change of inherited traits over successive generations [13]. This definition is based upon the theory of evolution by natural selection as presented by Charles Darwin [15]. In natural selection, inherited traits which facilitate the survival of a species become more common throughout a population [21]. More specifically, inherited traits may lead to adaptation when a trait produces a positive structural or behavioral characteristic relative to an environment [24]. An example diagram of adaptation is shown in Figure 3-1. In this example, a simple aquatic ecosystem is shown with four different species. At the top of the diagram, a new shark species is introduced which disturbs the ecosystem. Over a long period of time, the system adapts or changes relative to the new shark species as shown by the arrow labeled adaptation. In this new ecosystem, the turtle species failed to adapt. Hence, the turtle is now on the verge of extinction. Unlike the turtle species, the two fish species adapted new defense mechanisms such as poison and camouflage to defend against the shark.

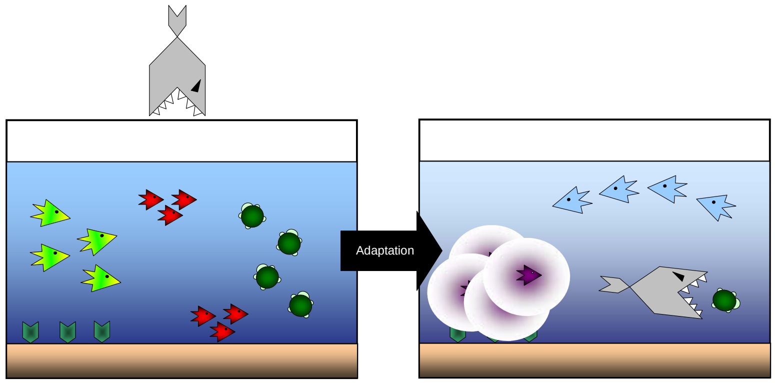


Figure 3-1: Adaptation in a biological system.

From this example, a basic overview of adaptation through evolution is presented, which by no means begins to explain the inherent complexity associated with biological adaptation. The diagram is used to illustrate a theory for the emergence of diversity and change within an ecosystem that provides survival benefits for certain species. The emergence of adaptation has been a popular debate in theoretical biology which has led to many great theories which account for the range of diversity observed in natural systems. One of these theories is that of co-adaptation often times referred to as co-evolution.

3.2 Co-evolution as a Mechanism for Evolution

Co-evolution is a phenomenon concerned with the mutual evolution between species [12]. The study of co-evolution began with Darwin's concept of evolution. In [15] Darwin states the following: "Wonderful and admirable as most instincts are, yet they cannot be considered as absolutely perfect: there is a constant struggle going on throughout nature between instinct of the one to escape its enemy and the other to secure its prey." This statement created a foundation for the study of co-evolution. Since then an enormous amount of work has been performed, resulting in a generalization of two types of co-evolution: a) competitive co-evolution and b) cooperative co-evolution.

Competitive co-evolution is concerned with the idea that no matter how well a species adapts to its environment, it must continually evolve to maintain its current fitness relative to other competing species. This succinct idea was first presented by Van Valen in his "Red Queen Hypothesis" [48]. In this work, Van Valen refers to this relationship as an evolutionary "arms race." This idea is addressed further by Dawkins and Krebs [16] in which the authors discuss the fate of evolution. In this discussion, the evolutionary arms race is argued to end in three possible states: a) extinction of one species, b) a stable state due to the location of a local optimum to prevent extinction, or c) the identification of a mutual local optimum or equilibrium. In this equilibrium, a state of cooperative co-evolution is born which is further discussed below.

Cooperative co-evolution can be explained from many different perspectives, but for this discussion the concept of mutualism is under investigation. With respect to mutualism, cooperative co-evolution pertains to the joint increase in fitness amongst two cooperating species, with fitness being a measure of survival or reproductive success [14]. Cooperation goes beyond merely the joint fitness increase due to the necessary dynamics which must be present for the aggregate behavior of cooperation. This necessity for properties which give rise to cooperation is stated by Axelrod [5] as follows: "…cooperation can evolve from small clusters of individuals who base their cooperation on reciprocity and have a small proportion of their interactions with each other." The critical word in that quotation is reciprocity. To articulate the idea of reciprocity, several examples are discussed below categorized into resource for resource, service for resource, and service for service cooperative reciprocity [14].

3.2.1 Resource – Resource

Resource for resource reciprocity refers to the relationship between two species; whereby, each species trades some form of resource for mutual benefit. This type of behavior can be shown in examples of arbuscular mycorrhiszal (AM) fungi-plant mutualism and legume-rhizobium mutualism.

In the fungi to plant mutualism, the AM fungus encourages plant growth by increasing access to soil resources such as nitrogenous compounds and water for the plant [14]. The plant provides carbohydrates required by the fungi for survival. In this relationship, each organism benefits through a resource to resource transfer in the form of soil availability for carbohydrates [7]. This fungal mutualism may be found on most plant species. Similar to this relationship, legumes tend to form mutualistic bonds with rhizobium for nitrogen fixation within root nodules.

In this relationship, legumes such as soybeans and rhizobium, a type of bacteria, provide each other with resources necessary for survival. Legumes require nitrogen from the soil to sustain their biological processes. To process this nitrogen, the legumes have formed a mutualistic relationship with rhizobium; whereby, rhizobium provides the legume with nitrogen in exchange for carbohydrates [25]. This represents another example of resource for resource reciprocity within a cooperative mutualistic relationship. Similar to this evolutionary phenomenon, service for resource mutualism is another type of cooperation between biological entities which emerged from cooperative co-evolution.

3.2.2 Service – Resource

Three examples of service for resource mutualism are provided. These examples are plants and their pollinators, human intestines and microorganisms, and humans and domesticated animals.

Plant and pollinator relationships represent a very common type of mutualism. In this type of relationship, a pollinator such as a bird or insect eats nectar or honey. In exchange for this food, a service is provided in which the pollen necessary for the plant's reproduction is dispersed. In this case, a resource (food) and service (pollen dispersal) represent the exchange which provides mutual benefit [14].

Yet another general example of resource for service is cleaner fish; cleaner fish come in a wide variety of species and provide a service to the host by removing parasites or dead skin or scales. The cleaner fish benefit from this relationship by obtaining food resources [11]. A similar interaction occurs in the human intestine with the interaction of a microorganism.

Human intestines contain a wide variety of microorganisms numbering in the hundreds of trillions. These microorganisms provide humans complex chemical transformations which the human body has not had to evolve itself. In this relationship, the microorganisms can communicate with the host, coordinate energy transformations, and provide critical chemical transformations [6]. In exchange, the human body provides the microorganisms with shelter and food. This represents yet another resource (food) and service (energy maintenance and chemical transformations) which displays service for resource mutualism.

3.2.3 Service – Service

The third and final general category for reciprocal mutualistic behavior is service for service relationships. These types of relationships are typically very rare to find in nature and often times occur in combination with the aforementioned mutual relationships. The first example is of anemone fish and sea anemones.

In this relationship, anemone fish are protected by the anemone's tentacles; which are usually lethal to other fish. In return the anemone fish protect sea anemones from butterfly fish which eat sea anemones. Furthermore, the anemone fish excrete ammonia which is used to feed the dinoflagellates which live on the anemones tentacles [41]. In this example, the service for service relationships is expressed solely by a mutual protection. This service for service relationship is also complemented by a resource component which is provided by the anemone fish in the form of ammonia for the symbiotic dinoflagellates. This mutual service based behavior is similar to the relationship expressed by some species of ants and trees.

It has been shown by Bronstein [10], that many ant species interact with plants for mutual benefits. For the ants, protection and shelter are provided by the plant. For the plant, protection is provided by the ants through warding off potential predators. In this relationship the primary driver behind the mutualism is protection. This represents a very similar relationship to the anemone fish and sea anemones.

3.2.4 Summarizing Co-evolution

From this brief discussion, two types of evolution perspectives are given in the form of competitive and cooperative co-evolution. The focus of this thesis is on cooperative co-evolution; therefore, several natural systems were used to articulate the idea of reciprocity in mutualistic behavior. From these examples, three types of relationships were identified which pertain to resource for resource, service for resource, and service for service reciprocity.

As a final example of cooperative co-evolution in the context of the ecosystem in Figure 3-1, cooperation is depicted as a trait which has evolved relative to the environmental disturbance of a new species entering an ecosystem, depicted by the shark. Over a long period of time, the shark species and fish species reach a final equilibrium state in which the fish species evolve a cooperative mechanism to ward off their predator. Hence, adaptation has resulted in cooperative behavior which increases the survival rate of the fish species by the exchange of protective services as depicted by the co-adaptation arrow. Through this cooperative behavior, the theory of co-evolution accounts for the interactions between species as a mechanism for adaptation.

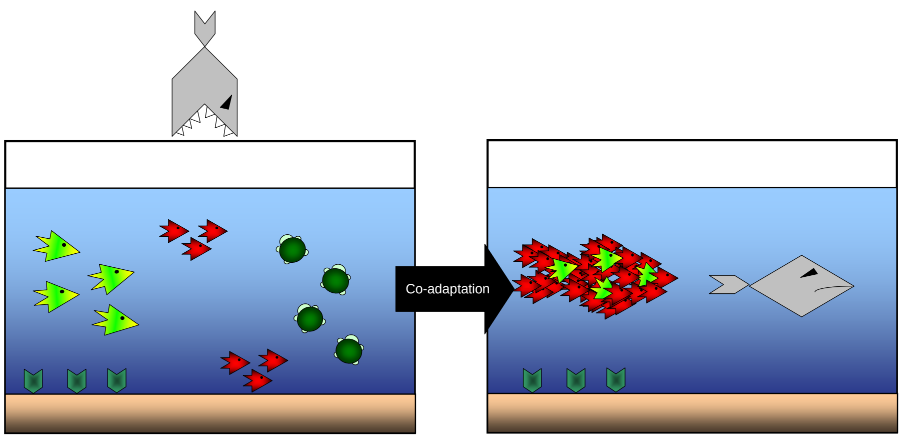


Figure 3-2: A biological example of co-evolution.

Once again this example by no means addresses all the issues associated with the true biological state of such an ecosystem. The concept is used to develop a running example of how adaptation and co-evolution might be extended to an evolving factory and more specifically reconfigurable manufacturing systems.

From this discussion and the running examples, this work is based upon resource for resource co-evolution; whereby, the resource for transfer is structural information to calculate reconfiguration cost. This exchange of topological information is described in the next section providing a walkthrough of how co-evolution is applied to an evolving factory and later reconfigurable manufacturing machines within that factory.

3.3 An Evolving Factory

As discussed in Chapter 1, a reconfigurable manufacturing system can be dealt with at two levels, the configuration level and the architecture level. At the configuration level, the system can adopt a configuration from the set of the total number of possible configurations. Change of configuration can be seen as strictly reconfiguration. The ability to reconfigure is limited by the architecture of the system as the total number of possible configurations is determined by the architecture of the system. In this thesis, the term ‘evolving’ is used to differentiate between a strictly reconfigurable system and a system in which the architecture changes. An evolving system is one in which the architecture of the system also changes when environmental conditions change. This section presents the concept of an evolving factory. In this thesis, the architecture of a factory is the number and type of components used to synthesize manufacturing machines. An example of an evolving factory is displayed in Figure 3-3.

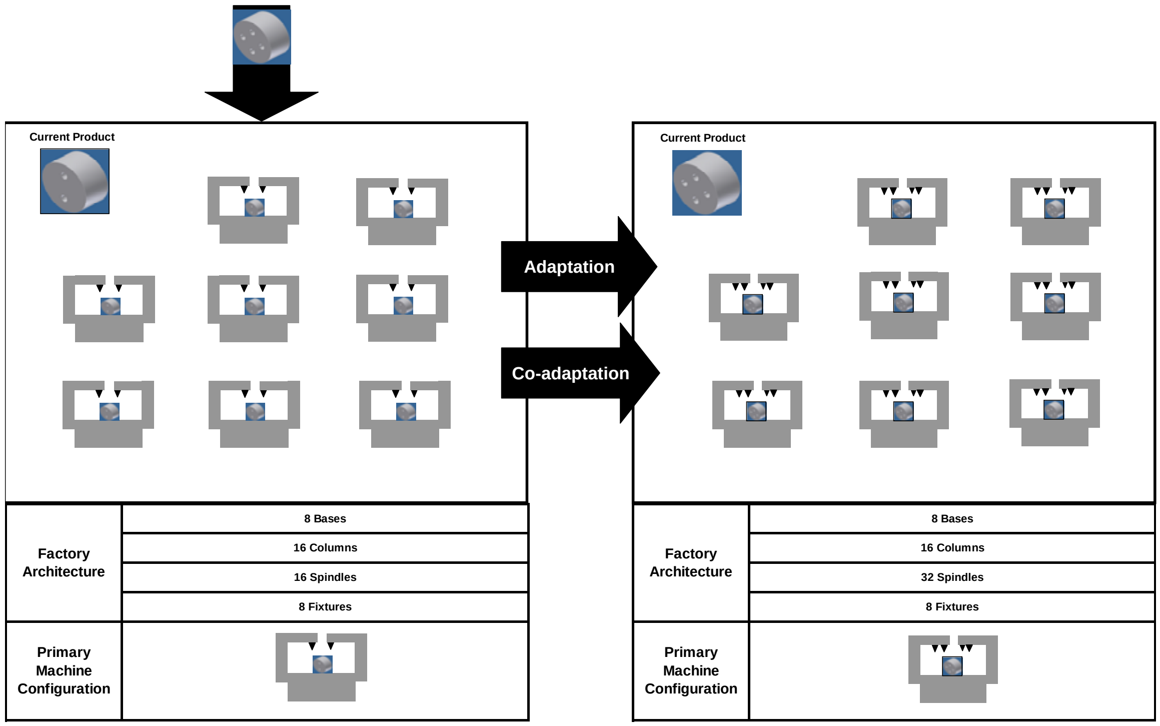


Figure 3-3: An evolving factory.

In this example, the factory may be viewed as an ecosystem. Hence, the environment for this factory would be the marketplace in which it operates. The possible disturbances which may arise in such a marketplace might be design or demand changes. In the example shown in Figure 3-3, the original factory is operating on a product featuring two holes. The factory, at this point, contains machines (species) that are arranged in specific configurations for the current product. These configurations are constructed by means of arranging and assembling machine components. The number of these components represents the architecture of the machine. The architecture of each machine includes one base, two columns, two spindles, and one fixture; hence, the architecture of the entire factory would be eight times the number of components in a single machine.

In the event of design and demand changes, the factory is disturbed which triggers evolution. The factory evolves to adapt to the environmental disturbance. During this evolution, the number and type of components within the factory change in response to the introduction of a new product. In Figure 3-3, the architectural adaptation results in doubling the number of spindles; hence, machine configurations are changed to better manufacture products. To enable adaptation, the machines synthesize their own architecture. This architectural adaptation involves the development of machines which are better suited for their environment; hence, the configuration can process a newly introduced product faster or more cost effectively. The example adaptation in Figure 3-3 is shown in Figure 3-4.

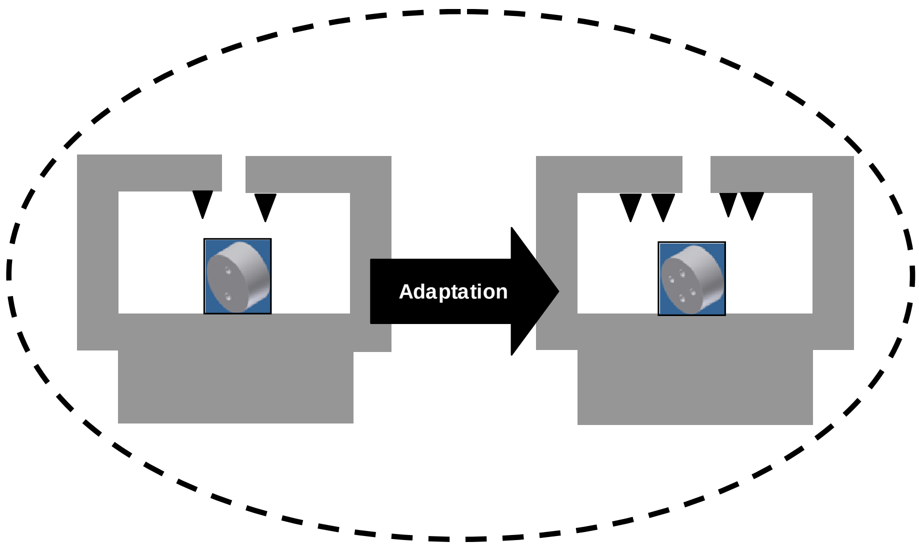


Figure 3-4: Example reconfiguration of two milling machines.

This circle represents a machine which has adapted its architecture and hence its configuration to process a different product. In this adaptation, the machine has developed two new spindles to enhance the processing capability of the machine architecture. Where there was once two spindles which machined two holes; there are four spindles which concurrently bore four holes resulting in a roughly equivalent machining time for a product with more features and higher volume removal. Hence from this minor difference in architecture, the machine is adapted to more effectively process the new product.

This simple example represents the primary topic of this thesis: architectural adaptation in reconfigurable manufacturing machines. More specifically, the focus of this thesis is the cooperative relationship which facilitates mutual adaptation between reconfigurable manufacturing machines. In the next section, an overview of the co-evolutionary method which embodies cooperation between reconfigurable manufacturing machines is presented.

3.4 Co-evolution as a Mechanism for an Evolving Factory

The concept of co-evolution is applied to determine the architecture of an evolving factory in this section. For simplicity, the discussion is based on the architecture of a single machine, i.e. the factory comprises of a single machine that has to be reconfigured for different products. The co-evolution approach will be based on the following context. The manufacturing company predicts that in the next year, it will produce three products. It needs to determine the architecture of a single machine that will be reconfigured to produce the three products at different times in the year. By applying the concept of co-evolution to an evolving factory, a method is developed to identify the architecture of a machine which provides similarity to the different configurations used to manufacture the three products. Hence, the architecture of the entire factory does not have to be more redundant than necessary. There is therefore a level of necessary flexibility within an evolving factory which is sufficient for operation. By achieving this level of flexibility through structural similarity, the required time to reconfigure a machine into another configuration would be minimized. The co-evolutionary approach is implemented based on multiple co-evolving computational agents. In this approach, each agent synthesizes the architecture of a machine for a product in the range of products it is to manufacture and cooperates with other agents which are synthesizing machines for other products to reduce reconfiguration cost. The rest of this section discusses the details of the method.

The co-evolution between multiple agents in the synthesis of the machine structure is shown in Figure 3-5 as the multi-agent system within the dotted lines. Each agent contains an evolutionary algorithm. The progressive solution development within the agent is defined by the inputs of product design and the availability of components in a component bank for the synthesis of acceptable reconfigurable manufacturing machines. This product design is characterized by features, volume to be removed, material, and batch size. The component bank is filled with machine components which characterize the range of components which are available to synthesize reconfigurable manufacturing machines. An agent is allocated for each product in the range of possible products to be manufactured to synthesize an appropriate manufacturing machine configuration for the product. All agents synthesize machine configurations in parallel from the same component bank. This component bank represents the range of components for the creation of desired machine configurations. Upon each iteration of the co-evolutionary algorithm, each agent calculates the cost of reconfiguring from its current configuration to the current configurations of all the other synthesized machines. This reconfiguration cost is used to update the fitness function of each agent. The inclusion of the reconfiguration cost would then alter the fitness of the current set of synthesized manufacturing machine structures. This process is depicted in Figure 3-5 as the exchange of information on the best synthesized machine configurations. This can be seen as the vertical arrows depicting information being sent on configuration A from agent A to agent B and information on configuration B being sent to agent A. This exchange of information is carried out between all agents i.e. agent A receives and sends information to all other agents. Accordingly, each agent then updates the evolutionary algorithm and synthesizes a new manufacturing machine.

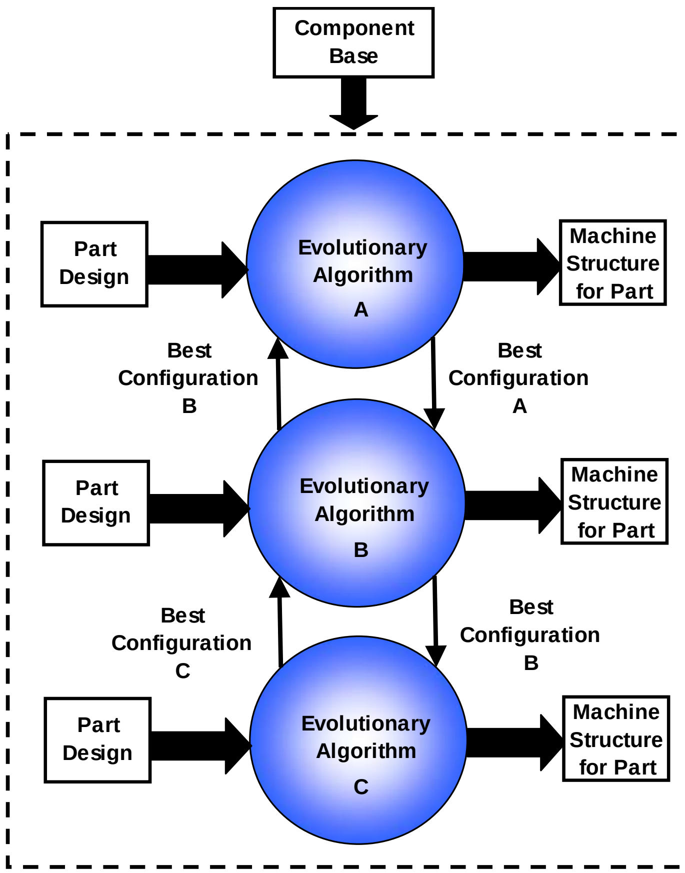


Figure 3-5: Co-evolving agent design synthesis.

The co-evolution of agents is continued till a termination criterion is fulfilled. The co-evolutionary algorithm can be terminated when a certain acceptable fitness value is reached for each agent. At the end of the co-evolutionary algorithm, a manufacturing machine configuration would have been identified for each product in the range of products to be manufactured. The synthesized machines would have been designed accounting for the tradeoffs between reconfiguring between configurations and the minimization of a single part cost. The final manufacturing machine would then reveal the components necessary for constructing each machine configuration. Upon removal of duplicated components between the various machine configurations, the necessary architecture for the reconfigurable manufacturing machine can be identified. This architecture represents set of components that will be used to construct the manufacturing machine when a change in the part design or quantity occurs.

3.5 Discussion

The co-evolutionary multi-agent design approach to reconfigurable manufacturing machine design has several salient features. Firstly, the approach allows for convertibility by minimizing the reconfiguration setup time between the machine configurations. In addition, the agent based structure of the algorithm allows reconfigurable manufacturing machines to be synthesized accounting for changes in the range of products which are to be manufactured by a company. For example, agents can be added or deleted depending on the projection of products to be manufactured based on market conditions. The architecture can then be altered according to market demands, creating an evolving factory.

From these salient features, the co-evolutionary multi-agent based approach has one primary characteristic which makes it unique amongst the other proposed reconfigurable system design methods: co-evolution. Incorporating co-evolution into the approach provides a means to synthesize machine configurations for adaptation to a changing product demand and evolving machine architecture. Hence, the machine architecture adapts as the product range evolves; thus, fulfilling the need for designing the changing architecture of reconfigurable machines within an evolving factory.

Using this concept of an evolving factory, a manufacturing company can then project the appropriate architecture necessary to machine a specific product range. Hence, the enterprise has the capability to adapt its factory's architecture to uncertainties associated with a dynamic and volatile market demand. These uncertainties include various product changes such as geometry or demand, economic developments, or unforeseen collaboration opportunities. The complexity associated with predicting and modeling this uncertainty is beyond the scope of this thesis. Therefore, uncertainty is excluded but may be incorporated in the future for better decision making. The focus of this thesis is on a synthesis approach for reconfigurable manufacturing machines.

In this approach, an automated method to designing the architecture of a reconfigurable manufacturing machine is presented. The components required to construct the machine configurations necessary to process a range of products may be identified. This set of components represents the necessary level of flexibility for an evolving factory to maintain sufficient operational capability relative to the predicted product range. In the event of a change in the product range, the approach is adaptable to a wider variety of product range uncertainties due to its network based approach. The network topology may be changed to reflect the order in which a product is demanded. The network can be formed in a linear fashion as shown previously in Figure 3-5 or in a parallel fashion when predictions cannot be sufficiently made. Therefore, the method can be adapted to accommodate a wide variety of demand uncertainty by adding or deleting agents from the network. The ability to change the topology of the network increases the flexibility of the method to be extended to a wide variety of product range predictions. Therefore, the method is capable of accommodating various needs and opportunities which arise in a typical manufacturing company; thus, addressing the research question: 'what should the architecture of a manufacturing machine be such that it can be reconfigured and adapted to changing needs and opportunities?'

Chapter 4 Application of Co-evolutionary Multi-Agent Design Method to the Design of Reconfigurable Milling Machine Architectures

In this chapter, the automated co-evolutionary multi-agent design method is applied to the design of reconfigurable milling machines (RMMs) architectures. In Section 4.1, the representation of solutions in the evolutionary algorithm is presented. In Section 4.2, the evaluation of solutions in the evolutionary algorithm is discussed. In Section 4.3, the co-evolutionary algorithm is presented and Section 4.4 concludes the chapter.

4.1 Solution Representation

For an evolutionary search algorithm, the solution representation is a critical characteristic which determines the structure and operation of the search algorithm. The solution representation defines the size of the search space and the nature in which that search space is explored. Typically, the solution representation of an evolutionary algorithm has a fixed length encoding which means that the number of design variables in the solution is fixed. This requires knowledge about the specific variables which constitute a satisfactory solution. In the context of design synthesis, the exact structure of the machine is unknown; therefore, a variable length solution representation is implemented.

The solution representation is based on establishing a hierarchy of RMM components. In this thesis, the function of the RMM is taken from the high level components of a milling machine. A diagram of these components is shown in Figure 4-1.

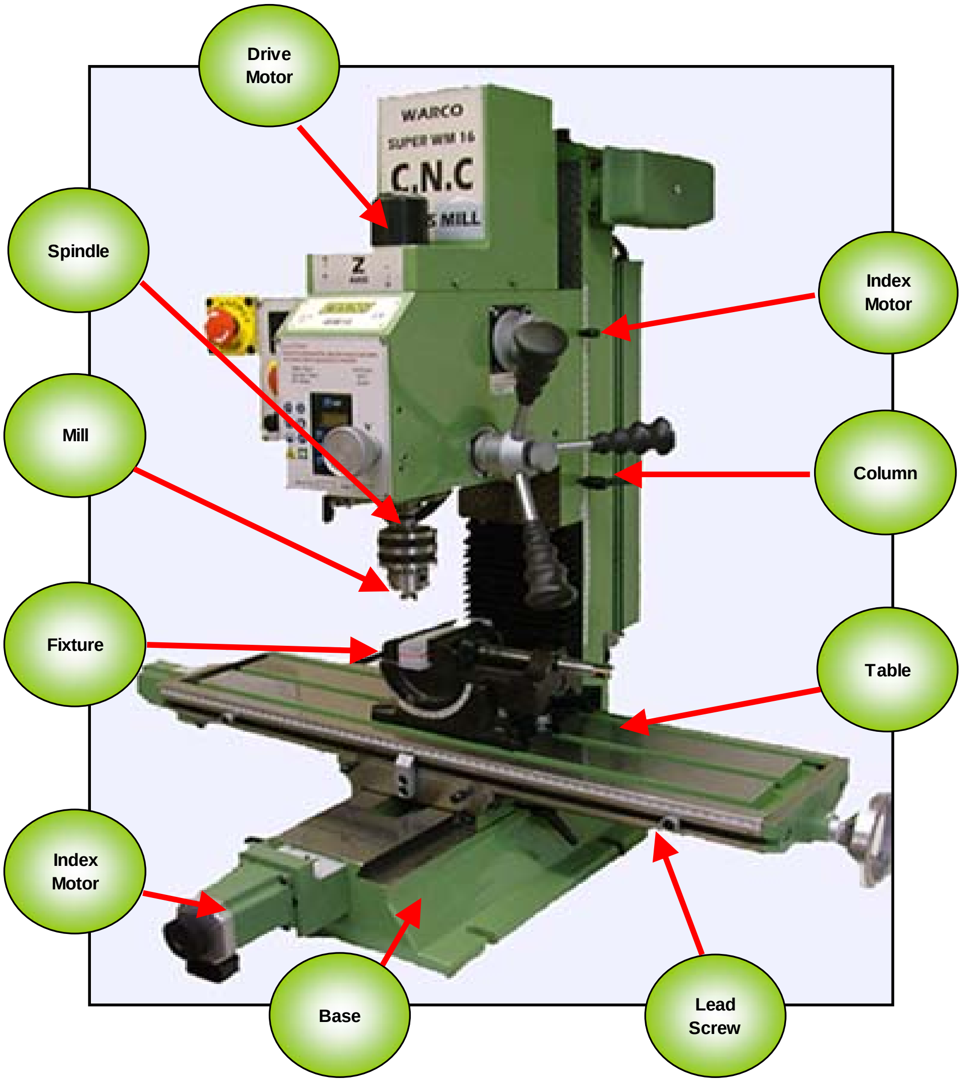


Figure 4-1: Machine components of the solution representation (adapted from [4]).

The RMM hierarchy is represented as shown in Figure 4-2. A RMM comprises of a single base structure and multiple possible columns to which functional units are attached. Two types of functional units can be connected to a base and column structure: a) tool holding and movement unit and b) work holding and movement unit.

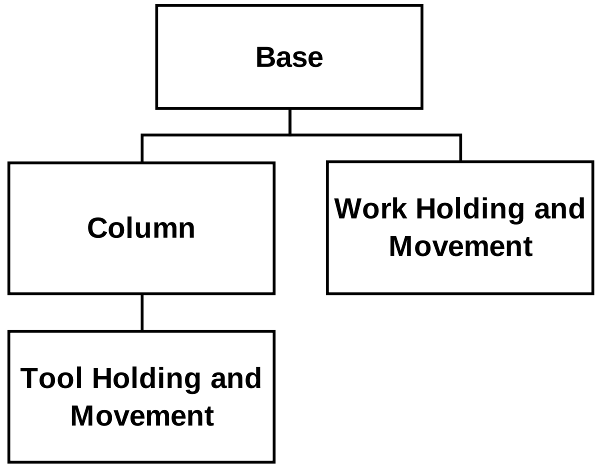


Figure 4-2: Reconfigurable milling machine representation.

A tool holding and movement functional unit is an assembly of components which provide translation and rotation to the tool. A tool holding and movement function unit is shown in Figure 4-3. This assembly of components is comprised of a spindle, drive motor, lead screw, indexing motor, and tool, as discussed in the following:

- Spindle – transmits rotation from the drive motor to the tool.
- Drive motor – provides rotation to the spindle.
- Lead screw and indexing motor – provides z axis translation to the tooling.

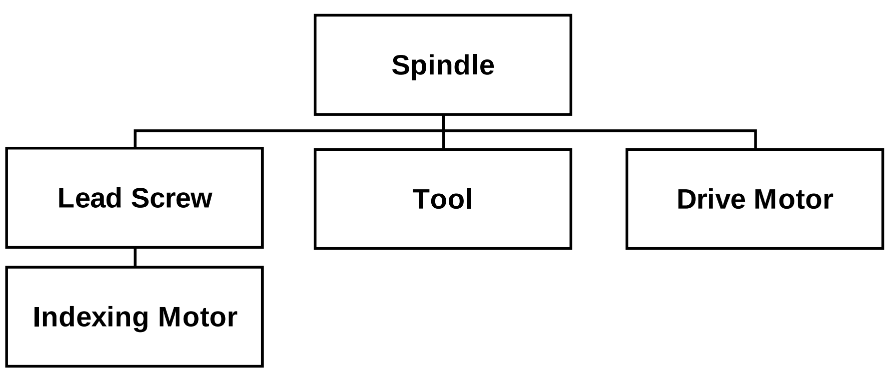


Figure 4-3: Tool holding and movement functional unit.

A work holding and movement functional unit is an assembly of components which provide support and translation to the workpiece. This assembly of components, shown in Figure 4-4, comprises of a table, fixture, two lead screws, and two indexing motors, discussed as follows:

- Table – transmits motion to the workpiece.
- Fixture – supports the workpiece during cutting.
- Two lead screws and indexing motors – provide x and y axis motion to workpiece.

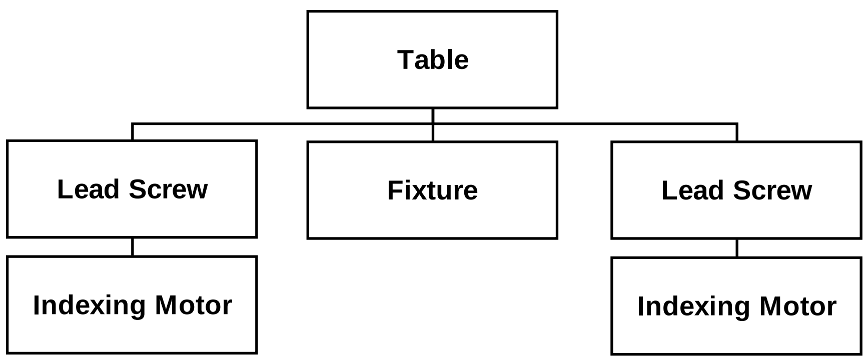


Figure 4-4: Work holding and movement functional unit.

Each solution is generated through the creation of a base and random numbers of columns. Random numbers of tool holding and movement and work holding and movement units are attached to the columns and base, respectively. An example solution representation is shown in Figure 4-5.

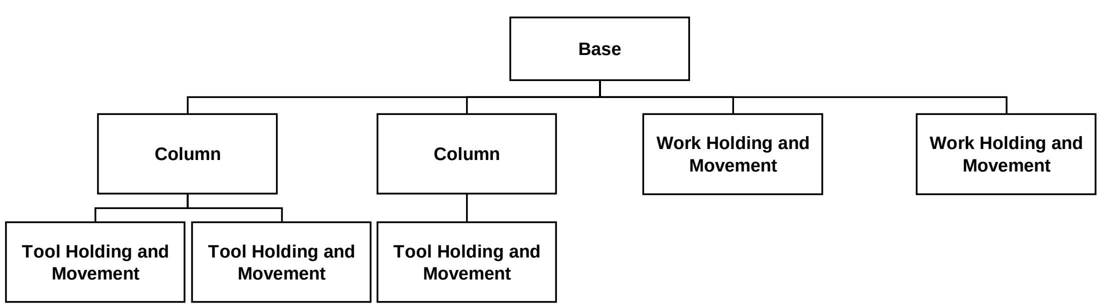


Figure 4-5: Example machine representation.

This example depicts a machine which has two columns with one and two tool holding and movement units. The three total tool holding units concurrently machine products located on the two work holding and movement units. Therefore, one work holding unit supports a product which is machined by two tool holding units.

4.2 Solution Evaluation

The characterization of a fitness function for a RMM must capture the most critical metrics for determining acceptable behavior of a system as it represents the means by which the machine configuration will be synthesized. In the developed algorithm, the fitness function includes metrics to evaluate the machining, capital, and reconfiguration cost between machine configurations. The quality of solution is evaluated by simulating the behavior of each milling machine configuration within the co-evolutionary design algorithm. Overall, the algorithm uses the behavior of the synthesized machines to simulate the cutting time and a comparison between system configurations to assign a fitness value. The fitness function is discussed below.

4.2.1 Fitness Function

The fitness represents the metric by which solutions are evaluated. In this case, the fitness is the average cost per part. The fitness function is formulated in the following way –

$$F = \frac{C_B + C_R}{B_S} + C_C$$

where C_B, C_R, C_C, and B_S are the machining cost per batch ($), cost to reconfigure ($), capital cost per piece ($) and the batch size, respectively. The details and assumptions associated with calculating these values are articulated in the following sections.

4.2.2 Machining Cost

Determining the machining cost involves two major calculations: a) batch processing time and b) the manufacturing cost per batch. To begin the determination of machining cost, batch processing time must be calculated. This calculation involves three sub routines which include a) inputting variables, b) a machinable feature check, and c) an iterative calculation of cutting time. This process is displayed in Figure 4-6.

To instantiate this process, the three subroutines must be sequentially accomplished to output the cutting time. To begin the process, the input variables must be supplied. In this implementation, the workpiece features are classified into three surface types: a) flat, b) cylindrical (only internal is considered), and c) irregular. Features are further classified into specific geometries such as flats, holes, and slots for end mill type cutters and t-slots and dove tails for face mill type cutters. The feature dimensions are modeled using a bounding box. For instance, a cylinder with a diameter of one inch and a length of 2 inches would be contained within a bounding box of 1 inch by 1 inch by 2 inches. The box is an overestimate, but appropriate for the level of fidelity in this model. After the features are specified, the batch size must be inputted.

The second process is a determination of the machinable features of a workpiece. This process is accomplished by scanning the geometry and type of every machine mill. If all of the features can be satisfied, the machine is deemed feasible. If all of the features cannot be machined, the machine is deemed infeasible and the number of machinable features is stored. Due to the machine infeasibility, a penalty function is instantiated on the final cutting time. The penalty function is discussed in step four of the cutting time determination as this is the point at which it is included.

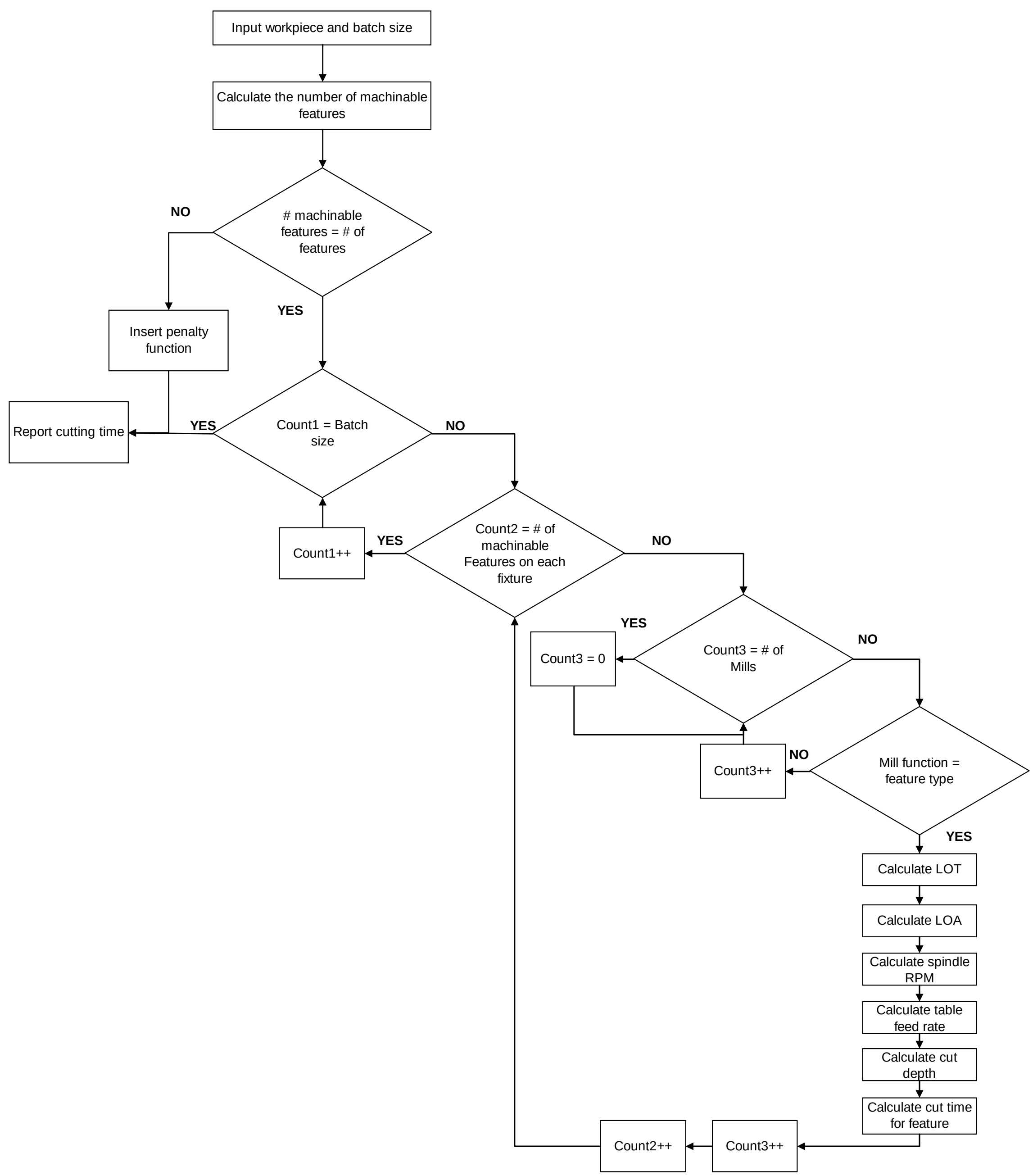


Figure 4-6: Procedure for calculating cut time.

The third and final routine in the cut time calculation is an iterative portion of the process which is comprised of four decision points and three processes. The decision points are a check for the count of processed parts, the satisfaction of the workpiece features, the machine mills, and mill function relative to the current feature. The three processes involve a scan of the workpiece features, a scan of the machine mills, and the cut time calculation. This process is explained in four steps which are to be iterated until the termination criterion is met. This process is performed in the following manner.

- First, a counter, initially assigned as zero, is checked for equality relative to the specified batch size. If this check is satisfied, the cutting time is reported; else the process enters the second step.
- The machinable workpiece features on each fixture are scanned and compared to the current mill. If a feature and mill are matched, then the algorithm enters step three else the mill count is incremented until an appropriate mill is located. The cutting assignment is explained in step 3 which is shown as calculate cut time for feature in Figure 4-6.
- To obtain the cutting time, another series of calculations are required which include estimating the spindle angular velocity, table feed rate, length of approach (LOA), length of over travel (LOT), single pass cut time, cut depth, number of required passes, and a cut time incorporating all of the passes required to machine a feature. This series of calculations is performed as follows.

The spindle RPM is used to determine the table feed rate and is given by [18] –

$$N_S = \frac{12V}{\pi D}$$

where V and D represent the cutting speed and cutter diameter, both of which are defined by the selected cutting tool. In this case, the cutting tool diameter is assumed to be in a range of acceptable diameters from 1/16 to 2 inches in increments of 1/16 inches. The cutting speed is a metric which is defined by the material and type of cutter. The data for cutting speed is taken from Table A-1 of [18] shown in Appendix A.

From the spindle RPM, the table feed rate can be calculated. The table feed rate determines the speed of workpiece translation. The feed rate is required to determine the machining operation cut time. The feed rate (inches per minute) of the table is given by [18] –

$$f_m = f_t N_S n$$

where f_t, N_S, and n represent the feed per tooth (inches per tooth), angular velocity of the spindle (rotations per minute (RPM)), and the number of teeth on the cutter (n), respectively. The mill variables, feed per tooth and number of cutter teeth, are dependent upon the interaction between the workpiece and tooling material. Mill materials are limited to carbide and high speed steel (HSS). For a listing of the workpiece materials, refer to Table A-1 [18] of Appendix A. The feed per tooth is also dependent upon the type of mill required. The data used for feed per tooth and the number of teeth may be found in Table A-1 [18] and Table A-2 [1] of Appendix A, respectively.

After the table feed rate is calculated, the length of approach (LOA) and length of over travel (LOT) must be calculated to determine single pass cut time. The LOA and LOT represent an addition of material to the cutting pass due to geometric features of the cutting tool. These geometric features of the cutting tool are different for various classes of mills such as vertical or horizontal mills. In this application, only vertical milling machines are synthesized; therefore, horizontal mills are not considered. For vertical mill types such as end or face mills, these cutting lengths may be calculated by the following [18] –

$$L_A = L_O = \sqrt{W\left(D - W\right)} \quad \text{for} \quad W < \frac{D}{2}$$
$$L_A = L_O = \frac{D}{2} \qquad \text{for} \quad W \geq \frac{D}{2}$$

where L_A, L_O, W, and D represent the length of approach (in.), length of over travel (in.), width of cut (in.), and cutter diameter (in.), respectively.

Once the spindle angular velocity, feed rate, LOA, and LOT are known, the single pass time for a feature is calculated. A single pass represents exactly one cut on a feature at the specified length of cut. This value may be calculated in the following manner [18] –

$$T_m = \frac{L + L_A + L_O}{f_m}$$

where Tm, L, L_A, and L_O represent the cutting time (minutes), length of cut (inches), length of approach (inches), and length of over travel (inches), respectively.

After the cutting time for a single pass is calculated, the maximum depth of cut must be calculated to determine the number of passes required to machine a feature. The cut depth is determined as follows [18] –

$$C_{D,\max} = \frac{\mu \cdot eff}{\mu_S f_m DOI}$$

where μ, μ_s, eff, f_m, and DOI represent the spindle drive motor output (hp), unit power (hp-min/cu. in.), efficiency of the spindle drive motor, table feed rate (ipm), and depth of immersion (in.), respectively. To determine these parameters, several assumptions are required. The motor output and efficiency are assumed to be five horse power at eighty percent efficiency [18]. Each mill is assumed to have a depth of immersion of 1.25 in. [18]. The unit power represents the required power needed at the spindle to remove a cubic inch of material [18]. Hence, the unit power is dependent upon the workpiece material. The table from which the unit power values were retrieved is shown in Table A-2 of Appendix A.

From this cut depth calculation, the required amount of passes for a cut can be calculated. The number of passes is determined by the following operation –

$$n_P = \left(\frac{f_d}{C_{D,\max}} \right) \left(\frac{f_w}{C_{W,\max}} \right)$$

where n_P, f_d, f_w, $C_{D,max}$, $C_{W,max}$ represent the number of passes, feature depth (in.), feature width (in.), max cut depth (in.), and max cut width (in.), respectively.

With the number of passes, the total cut time may be calculated by multiplying the single pass cut time by the number of passes to arrive at a final estimation for the time required to machine a feature. Once the total feature cut time is calculated, the value is stored with reference to the cutting mill. Then steps two and three are repeated until the machinable features of a workpiece have been cut. When all of the machinable features on a workpiece have been cut, the counter, denoted by count1 in Figure 4-6, is incremented to represent a machined workpiece. To machine all the workpieces, steps one through three must be repeated until the batch size has been met and all workpieces have been machined with reference to their machinable features.

- The final step involves the report of the final cutting time for the milling operation. If the number of machinable features equals the total number of features, the mill with the most machining time is reported as the total time required to process the batch. Else, if the number of machinable features is less than the total number of features, a penalty function is used to punish the machine's lack of capability. This operation is described below.

After the batch time has been estimated by the mill with the highest cutting time on the machine, the following equation is employed to punish infeasible machine configurations as they are undesirable, but necessary for searching the design space –

$$B_T = 300B(n_T - n)$$

where B_T, B, n_T, and n represent the final adjusted batch time (min.), calculated batch time (min.), number of total features, and number of machinable features. The addition of a constant value of 300 represents a multiplier determined by experience to dilate the solution in the event of a machine with a single mill that can satisfy one feature extremely well.

Another exception in the evaluation of infeasible solutions occurs in the event of a solution which cannot satisfy any features of a workpiece. To punish this infeasible solution, but maintain its diversity within the population of solutions, the batch processing time is assigned a value of ten times the product of the number of features and batch size. By characterizing the exception in this manner, the solution is scaled with the input variables such that the batch time increases with the demanded batch size and number of machinable features.

Once the exceptions to the penalty function are addressed, the batch time for infeasible configurations can be estimated. This allows the algorithm to consider these infeasible configurations for the possible good behaviors or components which they may provide. The next step in the solution evaluation includes the estimation of the cost per part which is discussed below.

After the cutting time for an entire batch is calculated, the total cost per batch may be estimated. This estimation involves the determination of machining and handling cost per piece. In the current model, tool life is neglected; therefore, tooling cost, and tool changing cost are ignored. The equations for the machining and handling cost are shown below –

$$C_O = C \cdot n_S$$

$$C_1 = (B_T + \frac{B_S}{2}) x C_O$$

$$C_4 = \frac{B_S}{2}$$

$$C_B = C_1 + C_4$$

where C_O, C_1, C_4, and C_B represent the operating cost, machining cost, nonproductive handling cost, and overall cost per batch, respectively. To solve the machining cost equations for the overall cost per batch, a few assumptions are made for the operating cost data. From [18], it is assumed that the operating cost (C) is sixty dollars per hour for a machine with a single spindle; therefore, an assumption is made that operating C cost would total one dollar per minute per spindle (n_s). Also, it is assumed that it requires one half of a minute to load and unload a single part. With this assumption, the machining cost (C_1) is estimated by multiplying the operating cost by the sum of the batch cutting time (B_T) and the time required to transfer the processed parts denoted by the batch size (B_S) divided by two. Furthermore, it is assumed that it takes approximately one half dollar per part for nonproductive handling cost (C_4) [18]. Therefore, it is possible to calculate the total cost per batch (C_B) by adding the machining cost and non productive handling cost. With a final value for the total cost per batch, an estimation of the reconfiguration cost can be determined. This calculation is explained in the next section.

4.2.3 Reconfiguration Cost

To characterize the reconfiguration setup cost, the configuration of a machine is compared to the next configuration it is required to assume. Through this comparison, a difference in machine components is revealed which identifies the required architectural adaptation of the machine. Once the difference in machine components is determined, assumptions are made to arrive at a final estimation for the cost required to reconfigure the machine. The equation for machine reconfiguration cost is shown below –

$$d_T = |n_{AT} - n_{BT}|$$

$$d_C = |n_{AC} - n_{BC}|$$

$$d_S = |n_{AS} - n_{BS}|$$

$$C_R = C_W\left(t_T d_T + t_C d_C + t_S d_S\right)$$

In this equation, the reconfiguration cost is modeled as the absolute value of the difference between the number of components (columns, spindles, and tables) of two machine configurations denoted by d_T, d_C, and d_S. This represents the number of machine units which must be added or subtracted to reconfigure to the next machine configuration. Thus, a perfect reconfiguration would be that of zero which would represent no additional setup.

To estimate the cost associated with setup time, assumptions are made for the time required for installation or disassembly (t_T, t_C, and t_S) and the average worker wage (C_W). The assumed time for installation or disassembly of a table, column, or spindle are two hours, three hours, and one hour, respectively. The assumed average worker wage is approximated as \$15 per hour. Hence, the total cost required to reconfigure a machine is estimated by the labor cost required to reconfigure the machine. After reconfiguration cost is estimated, the final variable in the fitness function can be calculated. This variable is the capital cost and is explained in the following section.

4.2.4 Capital Cost

The capital cost per piece is calculated by accounting for the number of different machine components (n), an assumed cost of each component (mc_n), the machine component types (mt_n), and an assumed number of processed workpieces over the entire lifecycle of the machine (L). The capital cost per piece is expressed as follows –

$$C_C = \frac{\sum_{i=1}^{n} mc_n mt_n}{L}$$

where the sum of the component costs is divided by the total number of components processed by a machine over its entire life cycle. The number of different components is taken from the different milling machine components shown in Figure 4-1. The assumed machine component costs are shown in Table 4-1.

Table 4-1: Assumed machine component costs.

Component	Cost ($ x 1000)
Table	5
Lead Screw	5
L.S. Motor	10
Fixture	3
Column	3
Spindle	4
Spindle Motor	15

The lifecycle number of workpieces was assumed to be one million. This number represents the assumed number of workpieces which can be machined over the entire lifetime of a reconfigurable milling machine. With these assumptions, the total processing cost per piece can be calculated. The cost per batch is added to the reconfiguration cost and divided by the total batch size. The capital cost per part is then added to this value to arrive at the total processing cost per piece. Hence, the fitness function is fully characterized.

4.3 Synthesizing Machine Architecture using an Evolutionary Algorithm

In this section the application of the co-evolutionary multi agent algorithm to the design of the architecture of a reconfigurable milling machine is described. There are seven steps in the algorithm. These steps are discussed in the following sections in the order they appear in the algorithm. To begin the algorithm, the input variables are discussed.

4.3.1 Products and Batch Sizes

For each agent, the input variables are a list of features to be machined on a part, list of dimensions associated with each feature, the batch size for the part, and the workpiece material. Due to the nature of machining, the input variables are directly tied to the mill and cutting time for the machine configuration. Features, dimensions, and material determine the type of tooling required for the operation. The batch size directly controls the amount of processing time that is required for an entire batch of parts.

4.3.2 Initial Population

An initial population of random machine configurations is generated from the component bank shown in Figure 4-7. The component bank holds the types of machine components selected for the design of RMMs.

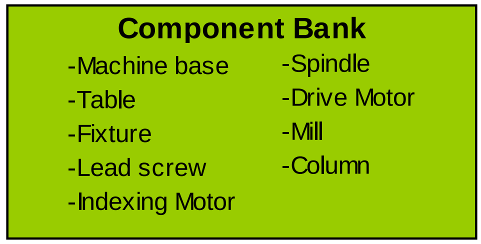


Figure 4-7: Milling machine component type bank.

The structure of the RMMs will therefore be synthesized from these basic components. To synthesize an initial machine configuration from this component bank, the following three steps are implemented to constrain the machine configuration:

- From the component bank, machines are synthesized by first instantiating a base structure.
- Next, random numbers of columns and work holding and movement units are attached to the base structure.
- To finish the initialization of machine configurations, random numbers of tool holding and movement units are added to each column.

An example machine configuration is shown in Figure 4-8. In this configuration, a base, column, tool holding and movement unit, and work holding and movement unit are shown. Like the configuration shown in Figure 4-8, the entire initial population of solutions is randomly created based upon the aforementioned synthesis constraints.

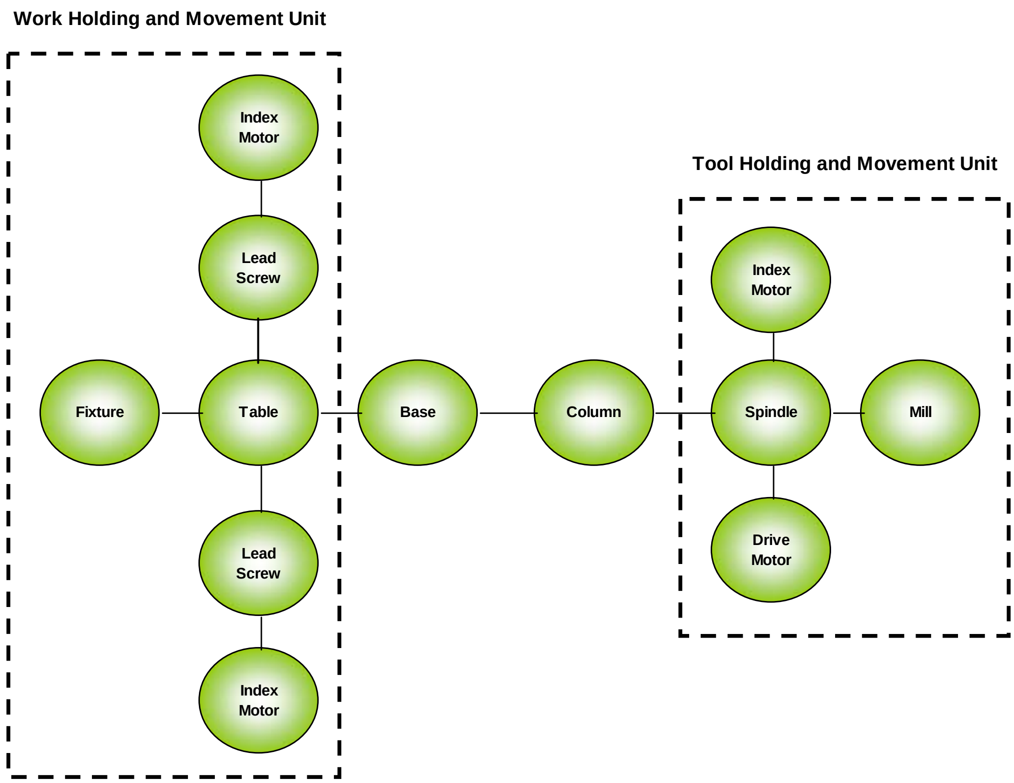


Figure 4-8: Example machine configuration.

4.3.3 Evaluation and Termination Check

Each solution in the initial population is then evaluated according to the fitness function and assigned a fitness value. A check for termination criterion is then performed. The termination criterion is set to one hundred generations. This criterion was determined by testing the algorithm to find the required time to generate acceptable machine configurations. When the selected number of generations is reached, the algorithm terminates and reports the best machine configurations.

4.3.4 Selection and Reproduction

If the termination criterion is not met, a new population of solutions is then formed by means of two selection methods. These selection methods are used to determine which solutions are retained for the next population. These two methods are tournament selection and elitist selection. The first selection method is tournament selection. Tournament selection is a proportionate selection method whereby solutions are sampled from the population and compared. More fit solutions are selected for reproduction. i.e. solutions with a higher fitness have a greater probability of being selected to be part of the new population. In elitist selection, the parent and child population are combined and sorted. The fittest half of the population of configurations from the combined population is retained. To produce a new population, the tournament selected group of solutions must be processed through evolutionary operators to introduce solution variation.

4.3.5 Evolutionary Operators

From this population pool, another new population of machine configurations is created through the use of evolutionary and topological operators. These operators are instantiated probabilistically. The evolutionary operators are shown in Figure 4-9. These operators include crossover and mutation.

Crossover is an operation involving an exchange of components between two candidate machines. In this design method, crossover has a 0.7 probability of being selected to generate new solutions. The crossover probability is subdivided into thirds by the different types of crossover. These crossover types include exchanging work holding units, columns, and tool holding units. An example of tool holding unit crossover is displayed in Figure 4-9. These types of crossover allow for the possible trade of structural components of two machine configurations. At the work holding level, crossover involves an exchange of the fixtures, tables, lead screws, and indexing motors between two candidate solutions. By allowing this trade, machines may increase or reduce their number of fixtures; thus, searching the design space with respect to an enhancement in parallel processing capability.

For column crossover, the candidate solutions trade entire assemblies of columns and tool holding units. This type of crossover grants the algorithm the ability to select entire tooling clusters to search for fitter configurations in larger increments through the search space. The larger increments in the search space provide more substantial additions or subtractions in terms of processing capability. This type of crossover is complemented by single tool holding unit crossover, which provides finer tuning of the configuration.

For tool holding unit crossover, as shown in Figure 4-9, the spindle, lead screw, spindle motors, and mills are traded between candidate solutions; thus, changing only one mill and introducing variation into the processing capability of a machine configuration. This operation is similar to the mutation operator which provides a random variation to the properties of a mill.

Mutation is the random selection of a new mill from the component bank to replace a pre-existing mill. The mutation probability is set at 0.2. When mutation occurs, any mill may be selected from the component bank to search the design space for a better component. In Figure 4-9, a mutation of an end mill results in a face mill.

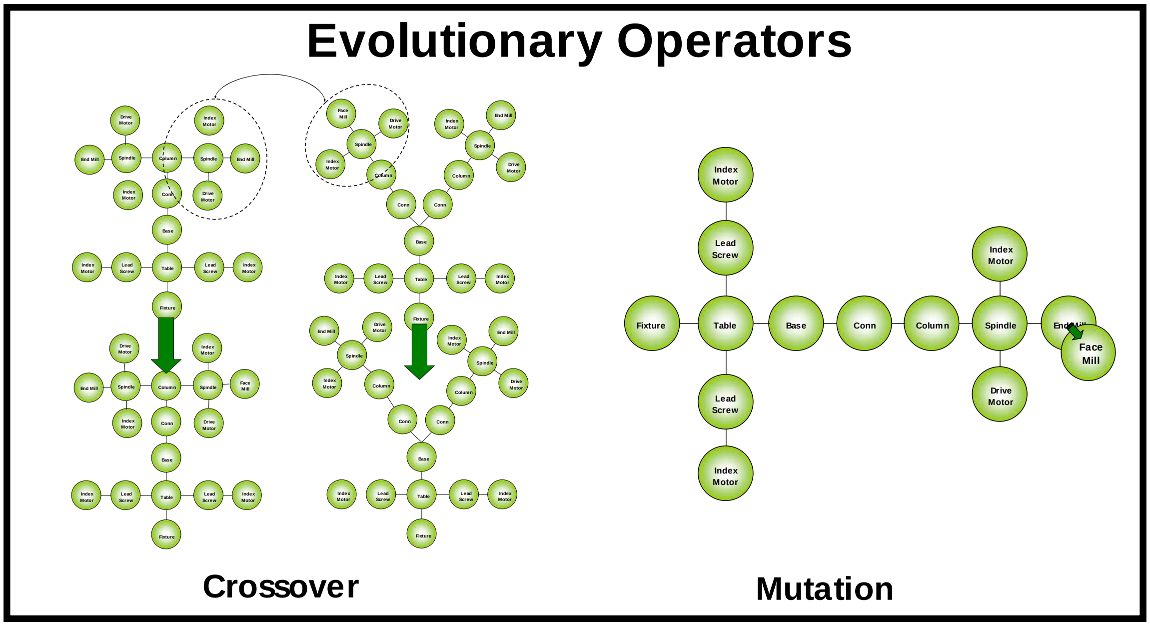


Figure 4-9: Evolutionary operators.

Another means of incorporating random change into the co-evolutionary algorithm is topological operators. Topological operators represent another form of mutation which changes the solution topology in various ways. A graphical representation of these operators is shown in Figure 4-10.

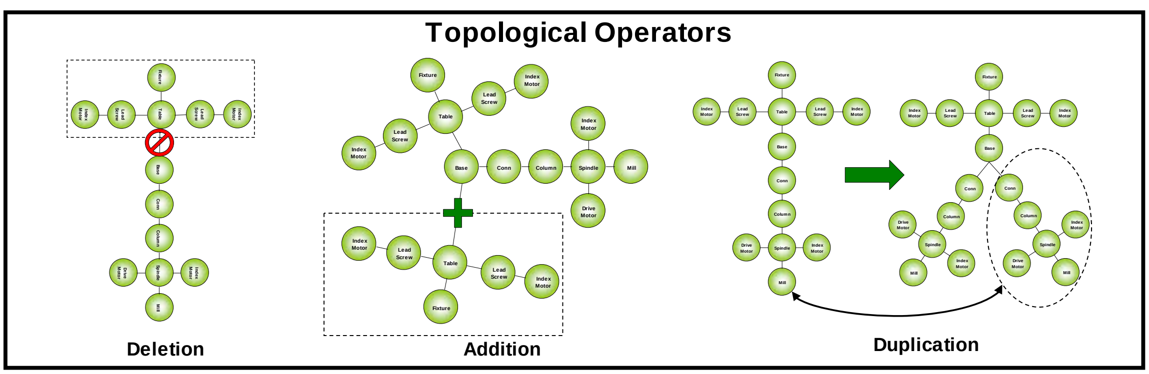


Figure 4-10: Topological operators.

In this implementation, a set of machine components may be added, deleted, or duplicated from a machine. Each topology operator has a 0.35 probability of occurrence. If deletion occurs, only work holding units, tool holding units, or columns may be removed. In Figure 4-10, a work holding unit is being deleted from the machine. For addition, random work holding units, tool holding units, or columns may be added to expand the topology of a solution. An example of this process is shown in Figure 4-10; whereby, a new tool holding unit is added. This operation is similar to the duplication procedure which also grows the configuration topology. The duplication operator may clone either a column assembly or an individual tool holding unit. When the operator clones a column of a machine, the algorithm copies a column and its tool holding units from a random machine onto a candidate machine configuration. In a similar fashion, the operator can copy an individual tool holding unit and install it on the configuration. Duplication marks the final type of topological operator which may be instantiated to create a new solution population. To continue in the algorithm, the individuals in the population are evaluated.

4.3.6 Transmitting Co-evolutionary Information and Evaluate

At this point, the co-evolution mechanism is triggered. From each agent within the co-evolutionary network, information on the fittest configuration is sent to other agents within the group of co-evolving agents. Each agent then calculates the reconfiguration cost and updates the fitness values.

4.3.7 Termination Check

Finally, the termination criterion is then checked again. If the criterion is not met, the algorithm repeats steps 4 through 7, as explained in Sections 4.3.3 through 4.3.7. The algorithm is finally terminated when the termination criterion is met. Upon termination, the algorithm reports each configuration for the products within the specified product range. From these machine configurations, an overall machine architecture may be derived which characterizes the necessary level of flexibility required to satisfactorily machine the projected set of products. Hence, the minimum number of machine components to meet the needs of the manufacturing enterprise is identified

4.4 Discussion and Limitations of Proposed Method

In this chapter, a co-evolving design method was presented and applied to the synthesis of RMMs. This design method focuses on identifying the appropriate architecture of a manufacturing machine relative to changing product needs. Hence within this design method, there are a few features which provide this unique capability in the context of the design synthesis of RMMs. These features include a graph-like solution representation based on function-behavior-form relationships, a co-evolutionary mechanism to derive an architectural plan for accommodating various product ranges, and a simulation for the coordination of n-spindles to concurrently machine products.

By using a graph based solution approach, the machine architecture can be synthesized from a feasible and non-feasible design space to produce a coherent machine. Furthermore, the graph based solution approach is computationally conducive for representing the co-evolution of machine architecture. Co-evolution compares the machine architectures to identify the appropriate machine architecture for a given product range. Once that machine architecture is synthesized, a concurrent machining simulation is implemented to evaluate the RMM. By evaluating the machines, the architecture solutions can be processed through an co-evolutionary search algorithm to identify the appropriate series of machine architectures for a give product range.

By incorporating these features into this design method, several good characteristics are granted such as the design synthesis of RMMs for adaptation to changing product needs. Also, the design method investigates the application of co-evolution to design synthesis. Despite the advantages associated with these features of the method, several limitations are presented as well to articulate the criticisms associated with such a design method. These limitations are presented in the following list:

- This method does not quantify the uncertainty of the product environment to measure extent of machine adaptation. By measuring the extent of adaptation, the limitations of the machine would be known.
- In this simulation model, a bounding box is used to represent product geometry. This provides a systematic overestimate of the required material removal. Since it is a systematic overestimate, all machines are synthesized based on the same feature assumption; therefore, this does not affect the end result of the synthesis algorithm with respect to identifying plausible machine architectures.
- Furthermore, the simulation does not account for geometric interference of the machine tools or tool path tracing. Therefore, the direction of machining is not considered. Hence, a regulatory mechanism is not present to determine how machine components should be arranged to access certain workpiece features. Also, the presence of tool path tracing and interference would grant a further mechanism to regulate the addition or deletion of columns.
- The operating cost, capital cost, and number of workpieces for a life cycle are assumed. The model behavior could be changed if these values were changed.

For instance, if fixtures were increased in cost, fewer fixtures could be added; thus, reducing the amount of concurrent machining within the machine architectures. Also, if the operating cost was changed, the synthesized machine architectures would necessarily change due to the increase or decrease in the number of spindles.

- Another assumption related to performance is the motor parameters. All drive motors are assumed to have five horsepower and eighty percent efficiency. If motor specification were included into the synthesis model, the motor output could change resulting in changing machine output costs. Along with this added fidelity, motor and spindle relationship constraints would be necessary to ensure that the spindle could tolerate the output torque of the motor.
- Finally, the fitness characterization and penalty function represent mathematical expressions of preferences associated with the desired behavior of this algorithm. By changing these preferences, the algorithm could be augmented to output different types of machine architectures.

In summary, a design method is presented with a tree based solution representation. To evaluate the solutions of this representation, a fitness function based on part cost was defined. After the fitness function was defined, the steps of the algorithm were explained in detail. These steps include variable inputs, population initialization, evaluation, selection and reproduction, evolutionary operators, information transmission, and termination check. The results from this design method are presented in the following chapter.

Chapter 5 Experiments and Results

In this chapter, the co-evolutionary multi-agent design method is validated by performing five experiments. This chapter first discusses the validation experiments, in Section 5.1. Section 5.2 presents the results from these experiments. In Section 5.3, a discussion of the experiments is presented in the context of validating the proposed hypothesis to conclude the chapter.

5.1 Introduction

The products used to study the design method of the reconfigurable milling machines are motor casings and automotive wheels. The motor casings represent a group of three products with identical features. The three automotive wheels have different features. The details associated with each product group are explained below.

5.1.1 Motor Casings

The motor casings are shown in Figure 5-1. The leftmost motor casing, denoted by the letter a), is the base motor. This component is machined from a blank of material with dimensions 3 inches by 1.5 inches by 1.5 inches. The two other motor casings, denoted by b) and c), are 2:1 and 3:1 scaled versions of the base motor.

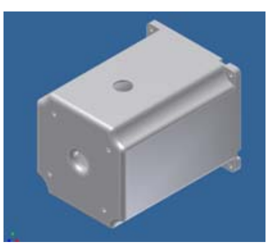
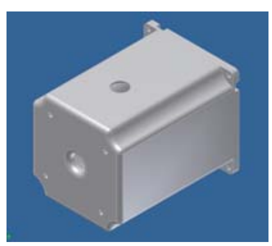
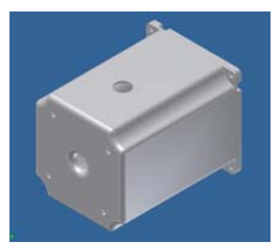
a)  b)  c) 

Figure 5-1: Motor casings: a) original b) 2 x scale c) 3 x scale

Each motor casing has nineteen features of identical type. These features are machined with milling operations. These features are summarized in Figure 5-2 with a general motor diagram. As mentioned before, each feature is assumed to be characterized by a bounding box of dimensions.

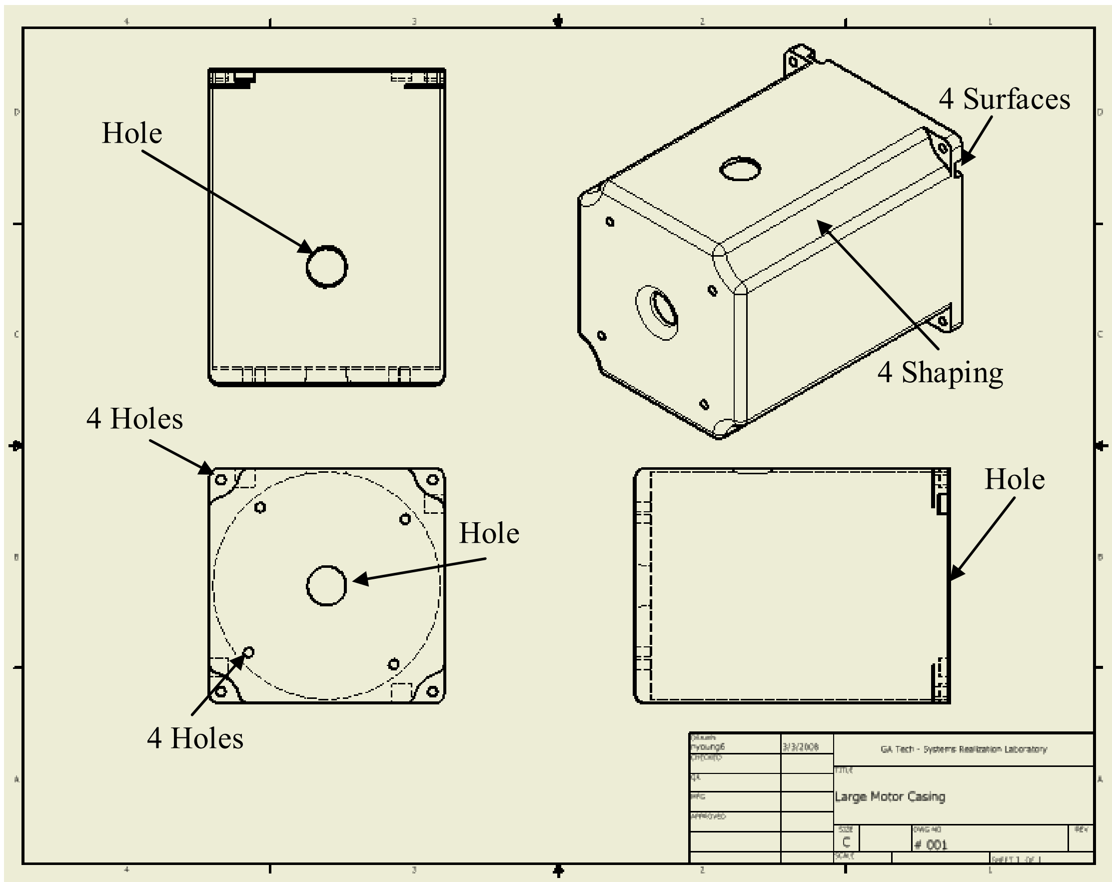


Figure 5-2: Motor casing feature drawing.

For instance, the hole on the front face of the small motor casing is denoted by the bounding box of a cylinder as represented by the dimensions 0.25 in., 0.25 in., and 0.1 in., respectively. For a full summary of the features and dimensions associated with casing a), b), and c) refer to Table 5-1 at the end of this section.

This set of workpieces is used to study the architecture of a reconfigurable milling machine relative to batch size variation, changing required volume removal, and workpiece material variation. To further study the structure of reconfigurable milling machines, a set of automotive wheels are introduced.

5.1.2 Automotive Wheels

The automotive wheels are shown in Figure 5-3. The wheels are each sixteen inches in diameter. The wheels range from five spoke to seven spoke resulting in a different number of features. The number of features on each wheel are nineteen, twenty one, and twenty three from left to right.

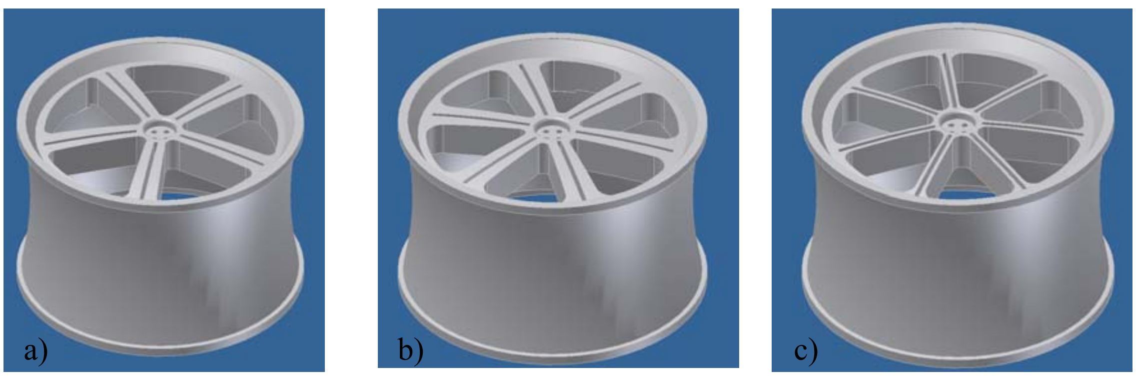


Figure 5-3: Automotive wheels: a) 5 spoke b) 6 spoke c) 7 spoke.

To machine these components, the features which would be typically milled are considered. These features include the triangular shapes, slots, surfaces, and holes. An example of the features on a six spoke wheel is shown in Figure 5-4. This example contains twenty one features. Therefore, the five spoke and seven spoke versions would necessarily have nineteen and twenty three features due to the decrease or increase in slots and triangular shapes.

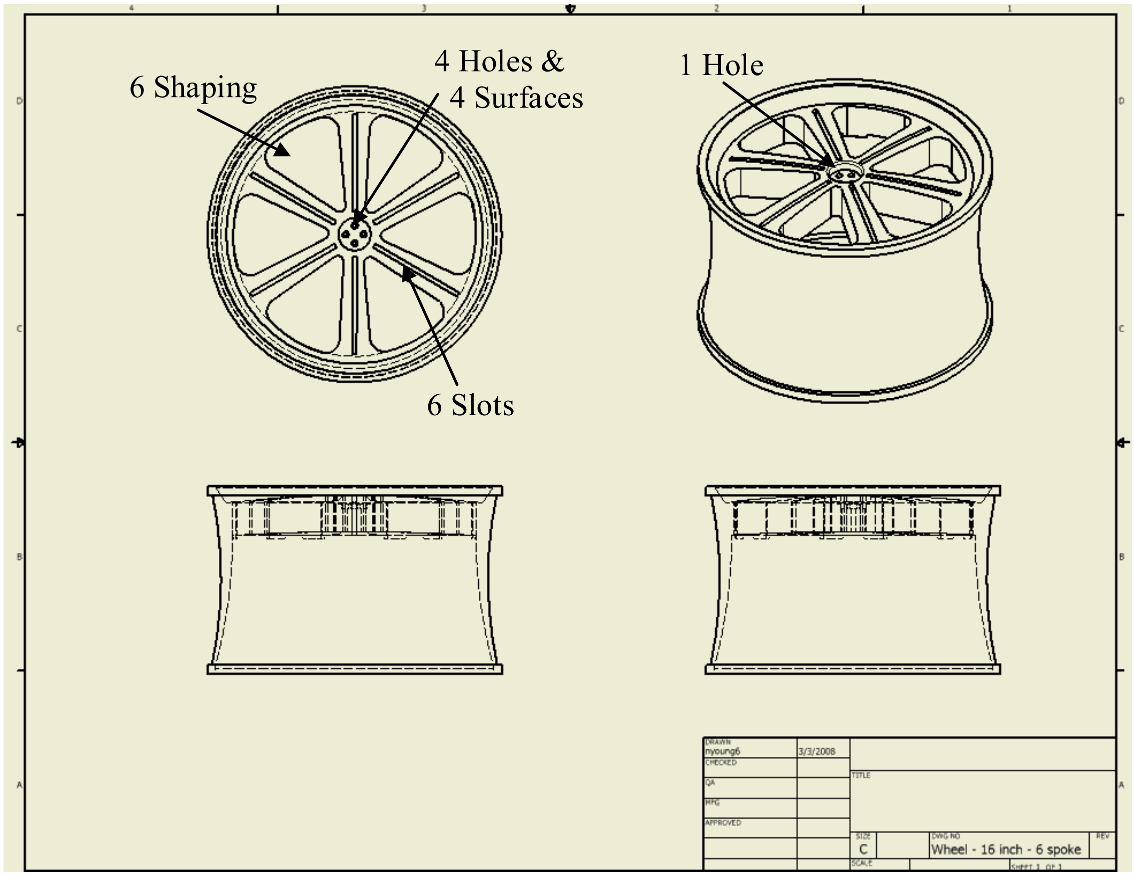


Figure 5-4: An example of the automotive wheel features.

Similar to the casings, these features are characterized by bounding boxes. Therefore, the machining operations are dependent upon the required amount of volume removal. A summary of the features and dimensions associated with each wheel is shown in Table 5-1.

This set of automotive wheels is used to study the architecture of a reconfigurable milling machine relative to a varied number of features. The experiments for these cases are explained in the next section.

Table 5-1: Summary of workpiece information.

	Feature	# Feat.	Width (in.)	Length (in.)	Depth (in.)	Vol. (cu.in.)	Tot. Vol. (cu. In.)	Tot. # Feat.
Small Motor Casing	Hole	1	1.45	1.45	2.90	6.10E+00	6.58	19
	Shaping	4	0.20	0.20	2.90	4.64E-01		
	Hole	2	0.25	0.25	0.10	1.25E-02		
	Holes	4	0.06	0.06	0.10	1.56E-03		
	Holes	4	0.06	0.03	0.10	6.63E-04		
	Surface	4	0.13	0.13	0.06	3.91E-03		
Medium Motor Casing	Hole	1	2.90	2.90	5.80	4.88E+01	52.64	19
	Shaping	4	0.40	0.40	5.80	3.71E+00		
	Hole	2	0.50	0.50	0.20	1.00E-01		
	Holes	4	0.13	0.13	0.20	1.25E-02		
	Holes	4	0.13	0.05	0.20	5.30E-03		
	Surface	4	0.25	0.25	0.13	3.13E-02		
Large Motor Casing	Hole	1	4.35	4.35	8.70	1.65E+02	177.66	19
	Shaping	4	0.60	0.60	8.70	1.25E+01		
	Hole	2	0.75	0.75	0.30	3.38E-01		
	Holes	4	0.19	0.19	0.30	4.22E-02		
	Holes	4	0.19	0.08	0.30	1.79E-02		
	Surface	4	0.38	0.38	0.19	1.05E-01		
16 Inch Wheel - 5 Spoke	Slots	5	0.25	5.50	2.00	1.38E+01	208.08	19
	Shaping	5	4.75	4.00	2.00	1.90E+02		
	Hole	1	2.00	2.00	1.00	4.00E+00		
	Holes	4	0.25	0.25	0.75	1.88E-01		
	Surfaces	4	0.38	0.38	0.25	1.41E-01		
16 Inch Wheel - 6 Spoke	Slots	6	0.25	5.50	2.00	1.65E+01	220.33	21
	Shaping	6	4.75	3.50	2.00	2.00E+02		
	Hole	1	2.00	2.00	1.00	4.00E+00		
	Holes	4	0.25	0.25	0.75	1.88E-01		
	Surfaces	4	0.38	0.38	0.25	1.41E-01		
16 Inch Wheel - 7 Spoke	Slots	7	0.25	5.50	2.00	1.93E+01	239.70	23
	Shaping	7	4.75	3.25	2.00	2.16E+02		
	Hole	1	2.00	2.00	1.00	4.00E+00		
	Holes	4	0.25	0.25	0.75	1.88E-01		
	Surfaces	4	0.38	0.38	0.25	1.41E-01		

5.1.3 Experiment Introduction

To study the relationships between workpiece variation and the architecture of reconfigurable machines, several experiments are presented. These experiments are performed on an assumed machine network. A diagram of this network structure is shown in Figure 5-5. This network represents the pathway of evolution a machine must take to accommodate three different product changes within an arbitrary time period. These experiments include varied batch sizes, varied volume removal, varied materials, a varied number of features, and changing reconfiguration cost.

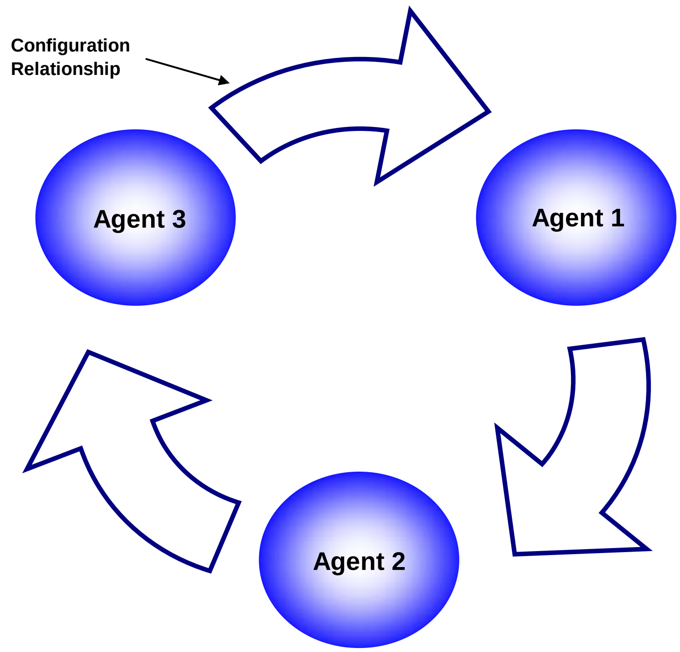


Figure 5-5: Assumed machine network relationships for selected experiments.

Experiment 1 – The purpose of experiment one is to investigate the relationship between batch size variation and the architectural requirements for a reconfigurable milling machine. In this experiment, three medium size aluminum motor casings are inputted to each machine. The batch size for agents A, B, and C are five thousand, ten thousand, and fifteen thousand. Thus, the scalability of the machine architecture is tested. A diagram of the experimental setup is shown in Figure 5-6.

From this experiment, the results include machine components, convergence graphs, and information pertaining to the change of machine architecture relative to batch size. The list of machine components includes the number of tables, lead screws, motors, fixtures, columns, spindles, and mills. Also, the part cost and fitness convergence graph are shown to demonstrate the performance of the algorithm.

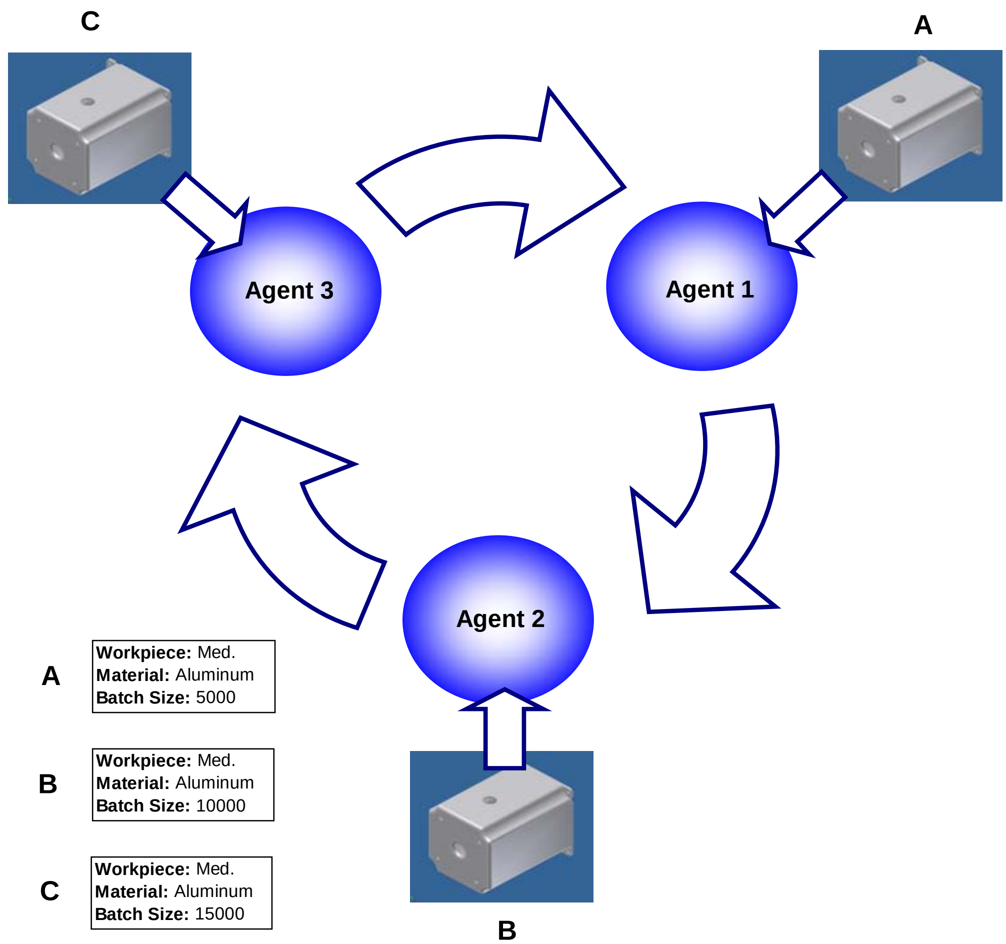


Figure 5-6: Experiment 1.

Experiment 2 – The purpose of experiment two is to investigate the relationship between machine architecture and varied volume removal. The individual products to each machine are varied relative to the amount of material that is required to be removed, as shown in Figure 5-7. Agent one, two, and three receive ten thousand small, medium, and large aluminum motor casings, respectively. This results in a machining system which must minimize the reconfiguration cost and cut time for each agent relative to differences in material removal rate. The volume removal required for part A, B, and C is 6.57, 52.59, and 177.49 cubic inches, respectively. The volume difference between A and B, B and C, and C and A are 45.72, 124.9, and 170.92 cubic inches.

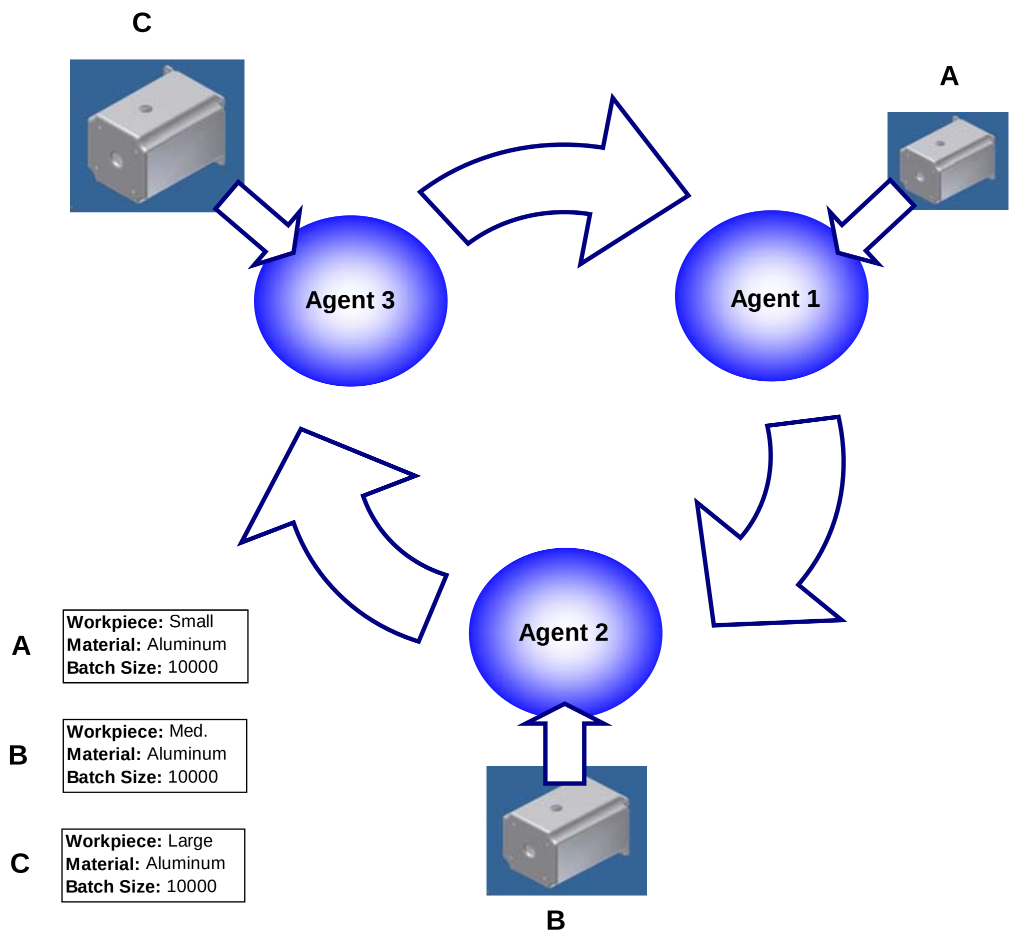


Figure 5-7: Experiment 2.

The results include similar information to experiment one. The primary difference is the focus on the varied volume removal requirements; therefore, information is presented which compares the system from a material volume perspective.

Experiment 3 – The purpose of experiment three is to investigate the relationship between machine architecture and material variation. By focusing on this aspect, the architecture between machines is compared in the context of material variation. This material variation is embodied by supplying different materials to each agent. The different material inputs are shown in Figure 5-8. Agent A, B, and C receive a fixed batch size of 10,000 medium motor casings with bronze, 100-150 BHN steel, and 350-450 BHN steel material inputs. These materials are selected for their various unit power and required cutting times. The specific numbers for these parameters are in Table A-1 and Table A-2 of Appendix A.

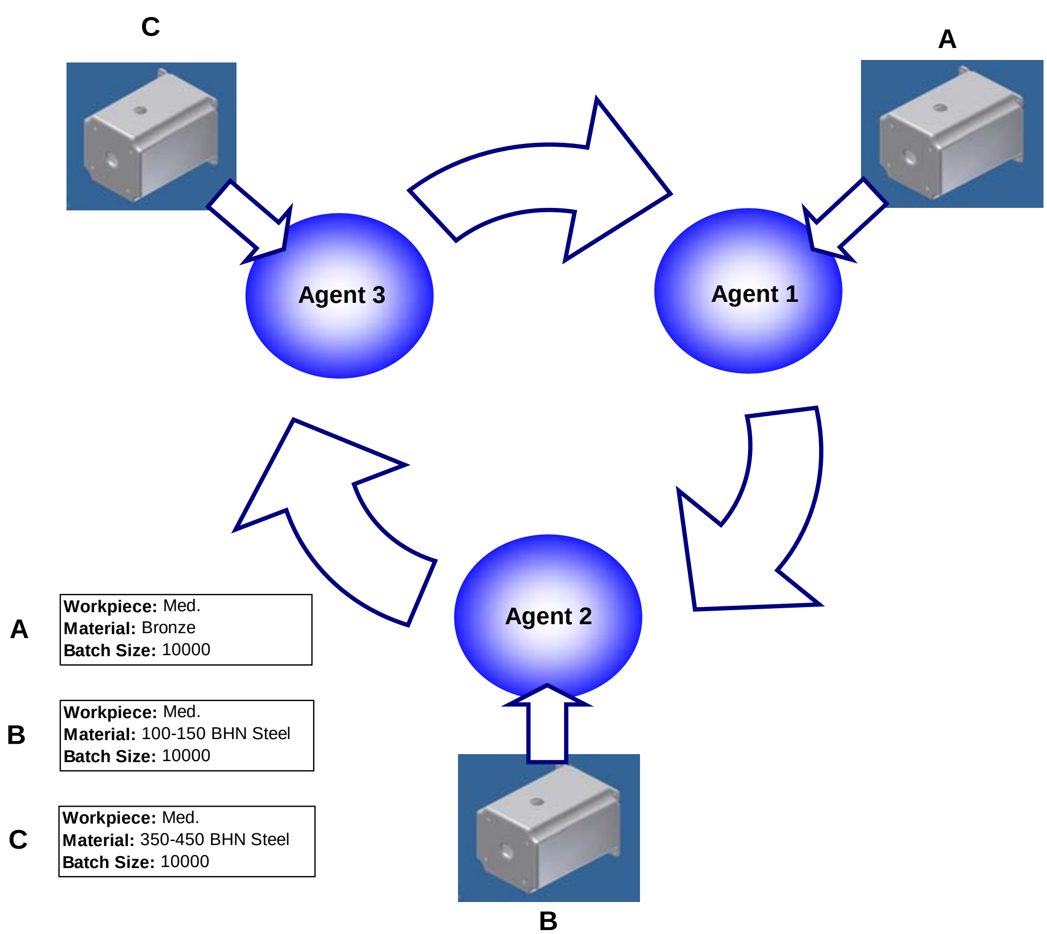


Figure 5-8: Experiment 3.

For experiment three, the results are formatted similar to that of experiment two. The primary difference is the focus on the varied materials for each agent. The information presented to represent the results contains data on the machine structure relative to material.

Experiment 4 – The purpose of experiment four is to investigate the evolution of machine architecture relative to changing product features. In this case, the products are changed to automotive wheels to introduce feature variation amongst a product range. The experimental setup is shown in Figure 5-9. Agent A, B, and C receive batches of five thousand five, six, and seven spoke aluminum wheels, respectively.

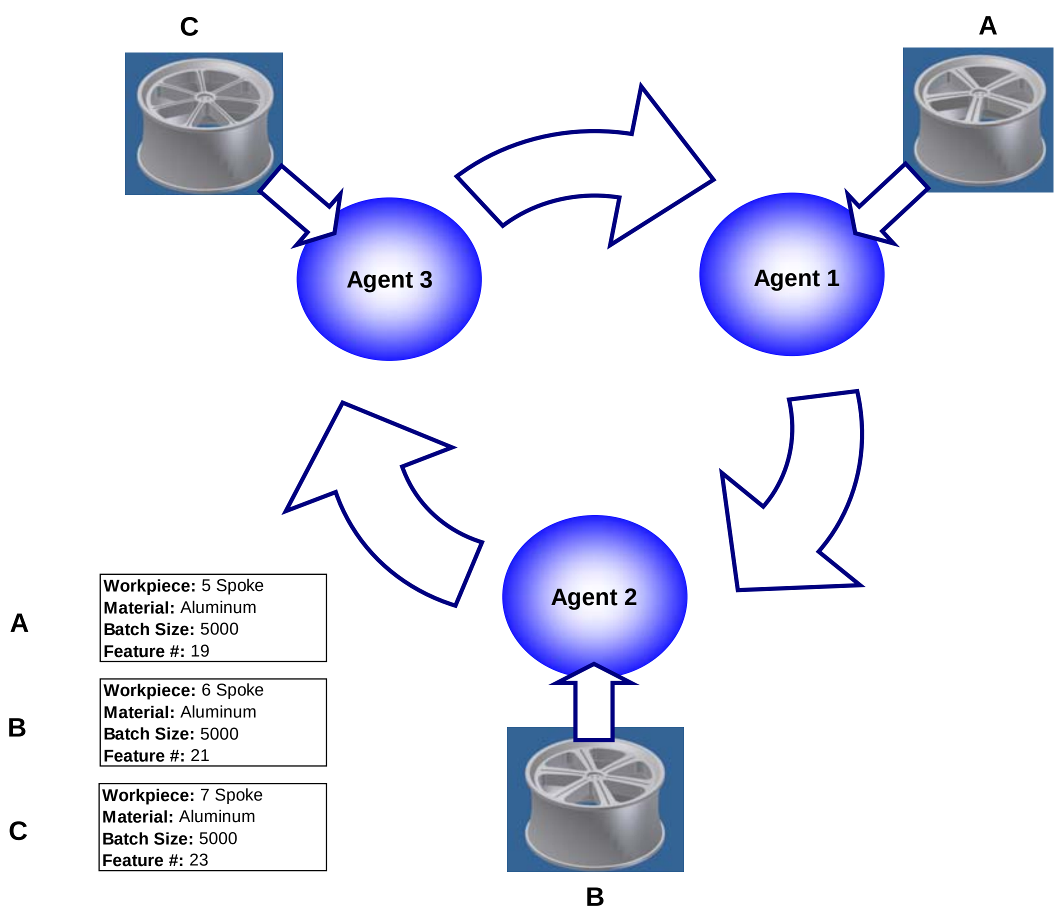


Figure 5-9: Experiment 4.

In this experiment, the results are presented in a similar fashion to the previous examples. The primary focus of this study is to present the required architectural evolution of the machine in reference to the number of features required to be machined.

Experiment 5 – The purpose of experiment five is to investigate changing reconfiguration cost relative to architectural change. In this case, the products are small motor casings. Small motor casings are used to reduce the amount of required volume removal; thus, increasing the reconfiguration cost relative to the overall part cost. The experimental setup is shown in Figure 5-10.

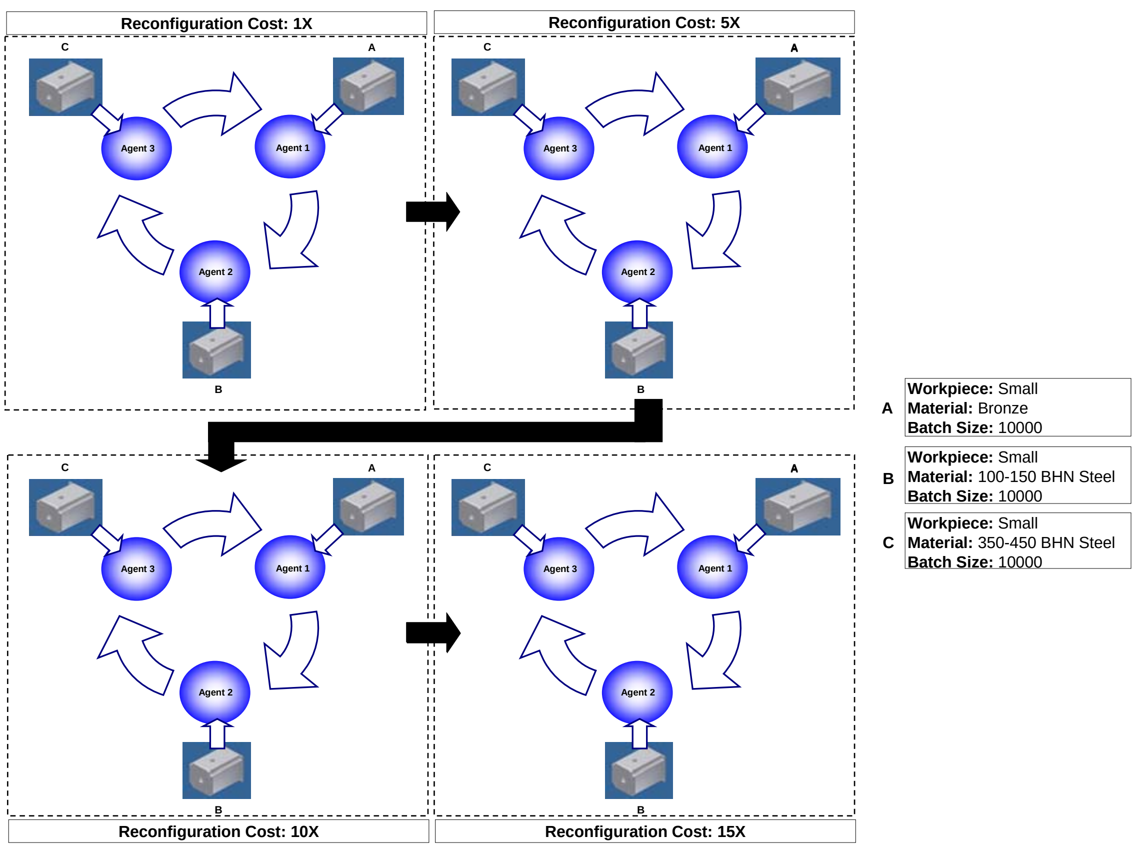


Figure 5-10: Experiment 5.

In this setup, four identical product ranges and machine networks are shown.. Within each machine network, Agent A, B, and C receive batches of ten thousand small motor casings. The material of the motor casings varies from bronze to soft steel, soft steel to hard steel, and hard steel to bronze. Material variation is used to introduce a large product change for driving a strong need to reconfigure. To study varying reconfiguration cost, the difference between each network is the reconfiguration cost which changes from one to fifteen.

In this experiment, the results will include each machine configuration. The part cost and amount of required reconfiguration will be presented relative to a varying reconfiguration cost.

5.2 Results

The results of the design method are presented in this section. To report the data for each experiment, a representative run is shown. The data for each experiment include the cost per part, reconfiguration cost, and machine architecture. After explaining the data from the experiment, the convergence graphs are discussed to validate the behavior of the selected approach.

5.2.1 Experiment 1

Experiment one is used to test the design approach relative to scalability. The selected batches include five thousand, ten thousand, and fifteen thousand medium motor casings with respect to each agent. The data for each agent is shown in Table 5-2. The corresponding cost per part for each machine configuration is \$10.4, \$6.8, and \$6.0. This decrease in part cost occurs for two reasons: a) capital cost and b) an adapted configuration.

Table 5-2: Experiment 1 machine data and architecture.

Machine Data			Machine Configuration								
Agent	Batch Size	Cost ($/part)	Base	Table	L.S.	I.M.	Fix.	Col.	Spin.	S.M.	Mills
1	5000	10.4	1	8	24	24	8	1	8	8	8
2	10000	6.8	1	1	9	9	1	1	7	7	7
3	15000	6	1	5	21	21	5	2	11	11	11
			Machine Architecture								
			1	8	24	24	8	2	11	11	11

The initial configuration has eight spindles and tables. Therefore, there is a spindle to concurrently machine on each table. This concurrent machining configuration drives the reconfiguration cost and capital cost upward resulting in a high cost per part. To adapt to a five thousand part increase in the batch size, the machine configuration evolves to a configuration with one work holding unit and seven tooling units. Therefore, the machine is capable of concurrently machining with seven spindles on a single workpiece. This configuration results in a $3.6 decrease in the part cost due to a decrease in the capital cost and reduction in machining cost. The reduction in capital cost is attributed to fewer required components to machine the new batch of products. Furthermore, the reduction in part cost is due to a smaller machining cost. This reduction in machining cost occurs due to the subtraction of a tooling unit on the machine configuration. By decreasing the number of tooling units, the operating cost of the machine is decreased resulting in a necessary drop in the part cost. As the machine adapts its configuration to new market conditions, the machine becomes better suited for machining products. Further evidence of positive adaptation is shown by the increase to fifteen thousand products. In this configuration change, the part cost is reduced by \$0.8. This reduction in cost is attributed to the more cost efficient dedicated architecture of the machine. This cost efficiency is attained by increasing the number of tooling and work holding units. By increasing these components, the machine configuration's capability is more dedicated for the current situation thus reducing the machining time.

From this identification of the necessary configurations for each batch size, a machine architecture is identified which can meet the requirements for scalability with a minimal amount of machine component redundancy. This architecture is shown in Table 5-2 and includes one base, two columns, eight work holding units, and eleven tool holding units. To further evaluate the behavior of the automated design method which created this machine architecture, the average fitness convergence is analyzed and displayed.

Further evidence of the decreasing part cost is presented in Figure 5-11. The average fitness represents the average part cost; therefore, the average fitness can be analyzed to study the behavior of the design method with respect to the current objectives. The first, second, and third agent average fitness are denoted by the red, blue, and green lines, respectively. As reflected by the part cost, the average fitness of the agent one population is significantly higher than the average fitness of both agent two and three. Furthermore, the agent two average fitness is higher than the average fitness of agent three.

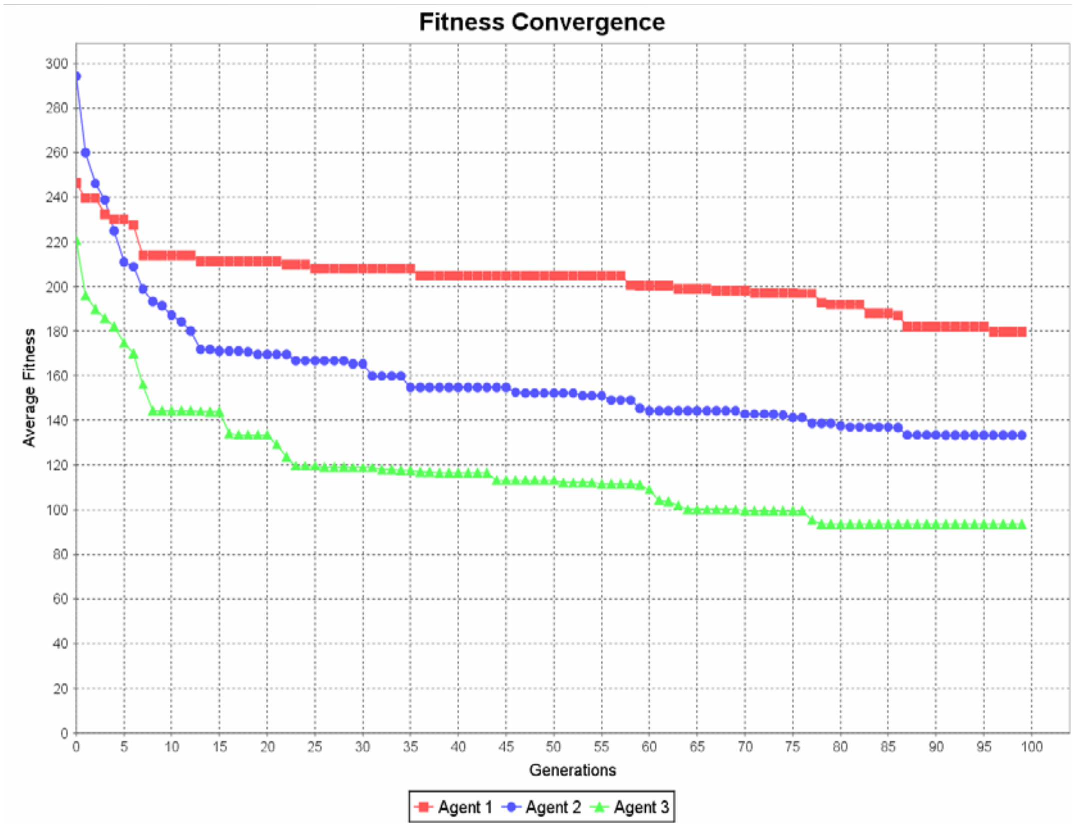


Figure 5-11: Experiment 1 fitness convergence.

5.2.2 Experiment 2

Experiment two is used to test the design approach relative to variable required volume removal. For each agent, 10,000 medium motor casings are studied with changing materials. These materials include bronze, soft steel, and hard steel The data for each agent is shown in Table 5-3. The corresponding cost per part for each machine configuration is \$1.6, \$11.6, and \$33.1.

Table 5-3: Experiment 2 machine data and architecture.

Machine Data			Machine Configuration								
Agent	Volume (cu.in.)	Cost ($/part)	Base	Table	L.S.	I.M.	Fix.	Col.	Spin.	S.M.	Mills
1	6.57	1.6	1	6	17	17	6	1	5	5	5
2	52.59	11.6	1	4	21	21	4	1	13	13	13
3	177.49	33.1	1	6	20	20	6	1	8	8	8
			Machine Architecture								
			1	6	21	21	6	1	13	13	13

The increase in the part cost is the result of progressively larger required material removal. The progressively larger material removal requirements results in more cutting passes which drastically increases the operating time and associated machining cost. To adapt to these drastic increases in material removal requirements, the machine configuration changes by manipulating the number of work holding and tooling units to accommodate the part cost and hence the capital, machining, and reconfiguration cost.

In response to the small motor casing, the initial part cost is $1.6. This part cost represents a compromise between the capital cost and the reconfiguration cost. Due to the features of small motor casings, three different mill types are required. Furthermore, five different tooling diameters are necessary. Therefore, the agent one configuration has the necessary level of tooling to sufficiently machine in parallel amongst six work holding units. The tool holding unit per part cost is $0.17; whereas, the reconfiguration cost for the tool holding unit is $0.015 for the current batch size. Therefore, a better agent one configuration is not found because the capital cost required to add spindles is more critical than the reconfiguration cost.

Unlike agent one, the agent two configuration contains many tooling units which represent a more dedicated machine configuration for the current medium motor casing. The second agent configuration includes four work holding units and thirteen tooling units. As with agent one, three mill types are required. The three mills can concurrently machine workpiece features on each of the fixtures. The remaining mill can rotate about the fixtures machining features to decrease the cutting time. Therefore, in this configuration, the necessity to machine in parallel overrides the capital cost and reconfiguration cost.

Unlike agent two, the agent three configuration has the same number of work holding units as agent one. Also, the agent three configuration has eight tooling units which represent a moderate number of spindles compared to agents one and two. The configuration has one tooling unit per work holding unit leaving two spare tooling units. As with configuration two, the spare tooling units rotate about the fixtures machining uncut features.

From the three agents, three configurations are identified for the current set of products. By examining these three configurations and subtracting redundant components, a final machine architecture is identified which represents the necessary level of flexibility for volume removal variation. This architecture includes six work holding units, thirteen tool holding units, one column, and a base. To ensure that the algorithm operated appropriately while locating this machine architecture, the convergence of the fitness is analyzed.

The fitness convergence for this experiment is shown in Figure 5-12. Similar to experiment one, the convergence may be used as a means to monitor the behavior of the automated design method. In this case, the agent three fitness is the largest followed by agent two and agent one. This ordering is produced by the progressively larger motor casings inputted into each agent.

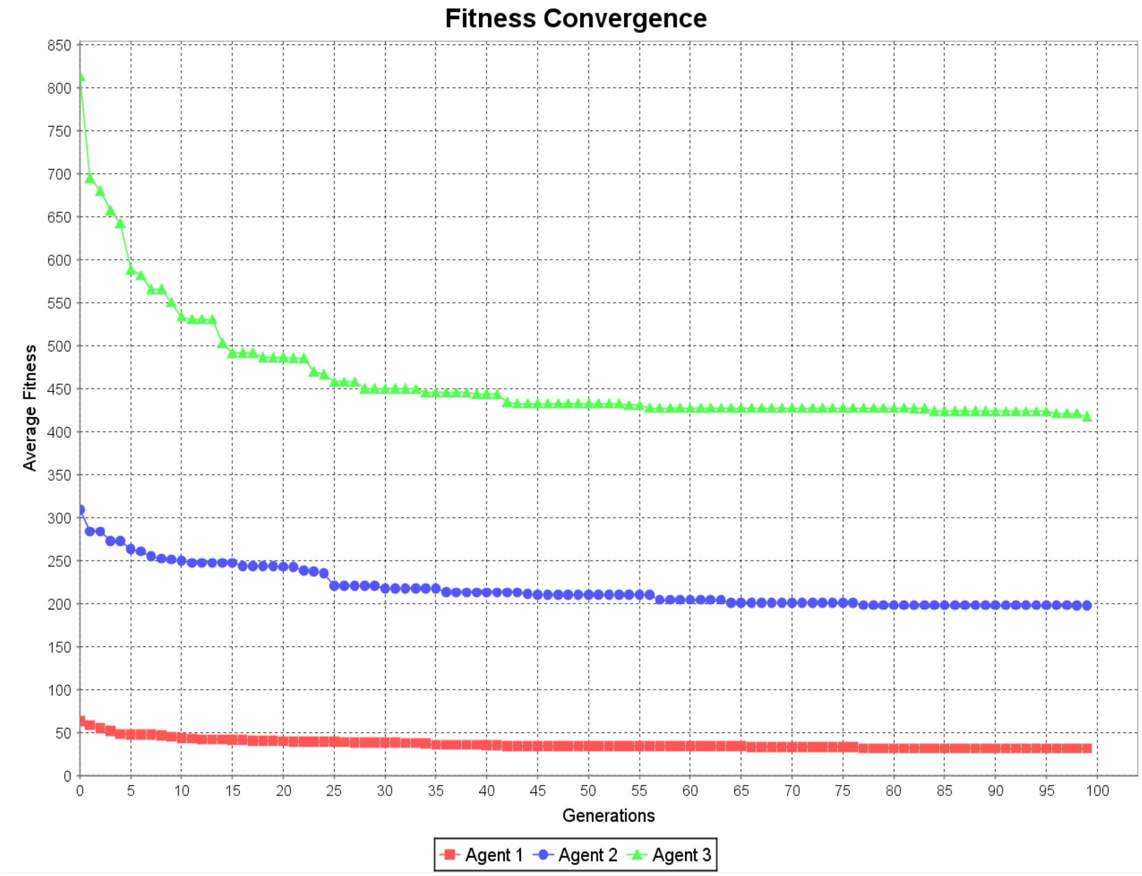


Figure 5-12: Experiment 2 fitness convergence.

5.2.3 Experiment 3

Experiment three tests the automated design method relative to material variation. For each agent, 10,000 medium motor casings are studied with materials ranging from low to high hardness values. The selected materials include bronze, soft steel, and hard steel. The data for each agent is shown in Table 5-4. The corresponding cost per part for each machine configuration is $7.4, $29.3, and $41.2.

Similar to experiment two, the part cost for three progressively increases for each subsequent agent. The part cost increase is the result of decreasing the cutting speed and increasing the unit power required for each material. The cutting speed controls the spindle angular velocity which in turn controls the feed rate. A change in the cutting speed changes the feed rate proportionally and affects the depth of cut. Furthermore, the depth of cut is reduced when the unit power is increased and vice versa.

Table 5-4: Experiment 3 machine data and architecture.

Machine Data			Machine Configuration								
Agent	Material	Cost ($/part)	Base	Table	L.S.	I.M.	Fix.	Col.	Spin.	S.M.	Mills
1	Bronze	7.4	1	1	5	5	1	1	3	3	3
2	100-150 BHN Steel	29.3	1	1	16	16	1	1	14	14	14
3	350-450 BHN Steel	41.2	1	1	9	9	1	1	7	7	7
			Machine Architecture								
			1	1	16	16	1	1	14	14	14

Due to the material properties, three machine configurations were synthesized beginning with agent one. Agent one is synthesized from the bronze medium motor casing. Agent one has the lowest cost per part. This low cost per part can be attributed to the minimalistic architecture the machine configuration requires. This architecture includes a base, one work holding unit, one column, and three tooling units. This represents the smallest number of components which can machine a motor casing. Therefore, the capital cost contributes minimally to the overall part cost.

Similar to agent one, agent two requires one base, fixture and column. Unlike agent one, agent two requires a significant number of tooling units to reach as satisfactory configuration. The satisfactory solution in this case represents a highly dedicated architecture in which there are fourteen spindles for nineteen features. Fourteen of the features can be machined in parallel. Consequently, the configuration has a substantial capitol cost and reconfiguration cost. The reconfiguration cost is due to the rather small architecture of configuration three.

Like agent two, agent three has one base, fixture, and column. Unlike agent two, the third agent has far fewer tooling units; hence, the capital cost is far smaller. While the capital cost is smaller, the overall part cost is much higher. To cope with this high cost, the resulting machine configuration contains half of the tooling units to reduce the operating cost of the machine. Since the configuration has a single fixture, the tooling units concurrently machine on a single workpiece.

From these configurations, an architecture is derived which is shown in Table 5-4. From this, the necessary level of flexibility to satisfy material variation is one base, one column, one work holding unit, and fourteen tool holding units. To gauge the behavior of the synthesis run, the fitness convergence is analyzed.

The fitness convergence for each machine configuration is shown in Figure 5-13. As with experiment two, the average part cost for each configuration decreases with the decrease in unit power and increase in cutting speed of the material. The algorithm's behavior is proven to be correct with reference to the expected behavior of the machine configuration relative to the preferences expressed in the fitness function.

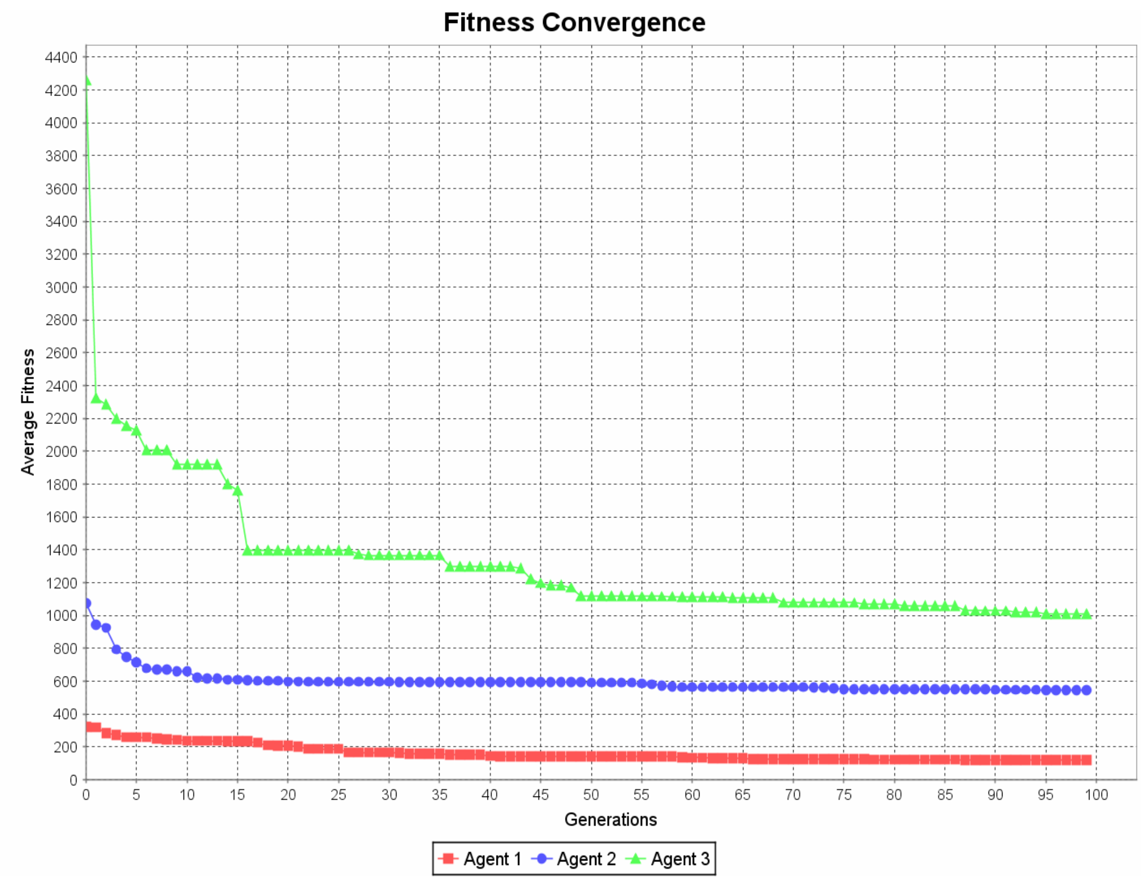


Figure 5-13: Experiment 3 fitness convergence.

5.2.4 Experiment 4

Experiment four tests the design method relative to a changing number of features. For each agent, 5,000 automotive wheels are studied. Each agent is supplied with a different class of automotive wheel. These classes include five, six, and seven spoke wheels. By increasing the number of spokes, the number of features increases along with the required amount of volume removal. The data for each agent is shown in Table 5-5. The corresponding cost per part for each machine configuration is \$10.8, \$13.2, and \$14.3.

Like both experiment two and three, the average part cost progressively increases as the number of features increases. This increase in part cost is caused by the increase in the required volume removal. The part cost increase can be further attributed to the varying architectures of each machine configuration.

Table 5-5: Experiment 4 machine data and architecture.

Machine Data			Machine Configuration								
Agent	Features	Cost ($/part)	Base	Table	L.S.	I.M.	Fix.	Col.	Spin.	S.M.	Mills
1	19	10.8	1	10	26	26	10	1	6	6	6
2	21	13.2	1	5	17	17	5	1	7	7	7
3	23	14.3	1	9	32	32	9	2	14	14	14
			Machine Architecture								
			1	10	32	32	10	2	14	14	14

The first machine configuration is synthesized from the five spoke wheel with the lowest required volume removal. The lowest required volume removal results in the lowest cost per part and fewest required tooling units. Furthermore, agent one requires the largest number of fixtures. This architecture can be explained by slow cutting on the five triangular features of the workpiece. The largest tooling diameter is two inches; therefore, the features require more than one pass to satisfy the required width of cut of 4.75 inches. Furthermore, the length and depth of the features is four and two inches, respectively. This results in a large required number of passes to fully machine the feature. While the large mills are making these machining passes, other mills are allowed to continuously work due to the large number of fixtures.

Agent two has far fewer fixtures than agent one while increasing the amount of required spindles by one. This architectural adaptation is credited to the increase in the number of features and required volume removal. The fewer number of work holding units promotes more concurrent machining on single workpieces. Due to the drastic decrease in work holding units and slight increase in tooling units, the capital cost is a much less significant factor in the cost increase of agent two. Conversely, the reconfiguration cost affects agent two greatly due to the large architecture of agent three.

Agent three requires one base, two columns, nine work holding units, and fourteen tooling units. Due to this architecture, the part cost is $14.3. This part cost is the result of the highly concurrent machining operation due to the large number of features on the workpiece. The larger number of features is conducive to concurrent machining operations; hence, there is a large number of work holding and tooling units. From this configuration, the capital and reconfiguration cost contribute significantly to the overall part cost relative to other configurations presented in this thesis.

From these configurations, an architecture is created which is shown in Table 5-4. From this, the required number of components to satisfy variation in the number of features is one base, two columns, ten work holding units, and fourteen tool holding units. To further analyze the behavior of the synthesis algorithm, the fitness convergence is analyzed.

The convergence plot is shown in Figure 5-14. The agent one, two, and three convergence data is denoted by red, blue, and green. As with the previous experiments, the average fitness progressively decreases as the number of features of the product increases. Therefore, the algorithm performs as expected with the increase in volume removal and the number of features.

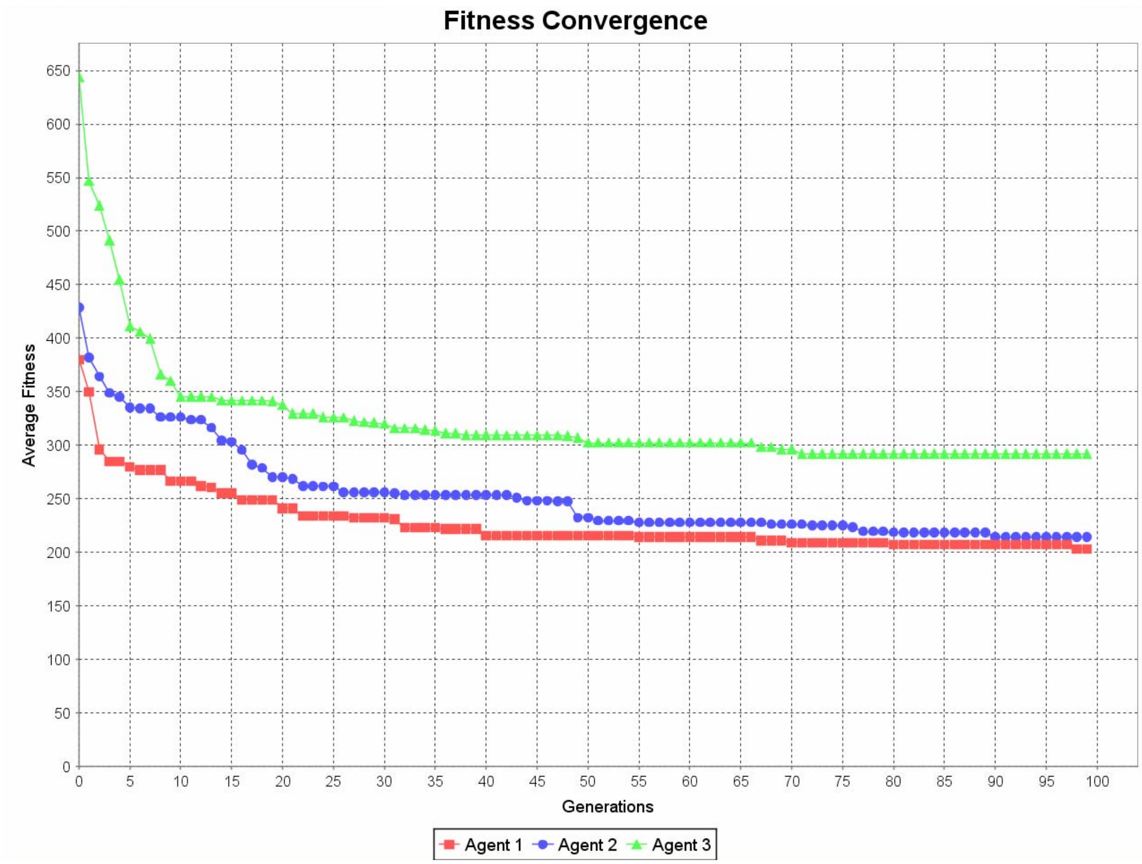


Figure 5-14: Experiment 4 fitness convergence.

5.2.5 Experiment 5

Experiment five tests the automated design method relative to changing the reconfiguration cost. For each agent, ten thousand small motor casings are studied. The materials are varied between bronze, soft steel, and hard steel. To study the reconfiguration cost, the cost of reconfiguration is multiplied by five, ten, and fifteen to observe the relationship between reconfiguration cost and machine architecture change. The results from this experiment are shown in Table 5-6. By increasing the reconfiguration cost for all components, a decreasing trend may be seen in the number of columns, work holding, and tooling units.

Table 5-6: Experiment 5 machine data and architecture.

Machine Data - Recon Cost 1x			Machine Configuration								
Agent	**Material**	**Cost ($/part)**	**Base**	**Table**	**L.S.**	**I.M.**	**Fix.**	**Col.**	**Spin.**	**S.M.**	**Mills**
1	Bronze	4	1	8	32	32	8	2	16	16	16
2	100-150 BHN Steel	8.2	1	3	17	17	3	1	11	11	11
3	350-450 BHN Steel	17.6	1	1	16	16	1	1	14	14	14
			Machine Architecture								
			1	8	32	32	8	2	16	16	16
Machine Data - Recon Cost 5x			**Machine Configuration**								
Agent	**Material**	**Cost ($/part)**	**Base**	**Table**	**L.S.**	**I.M.**	**Fix.**	**Col.**	**Spin.**	**S.M.**	**Mills**
1	Bronze	3.5	1	1	14	14	1	2	12	12	12
2	100-150 BHN Steel	9.4	1	2	13	13	2	2	9	9	9
3	350-450 BHN Steel	14.2	1	7	21	21	7	1	7	7	7
			Machine Architecture								
			1	7	21	21	7	2	12	12	12
Machine Data - Recon Cost 10x			**Machine Configuration**								
Agent	**Material**	**Cost ($/part)**	**Base**	**Table**	**L.S.**	**I.M.**	**Fix.**	**Col.**	**Spin.**	**S.M.**	**Mills**
1	Bronze	2.1	1	1	7	7	1	1	5	5	5
2	100-150 BHN Steel	6.7	1	6	21	21	6	2	9	9	9
3	350-450 BHN Steel	13.7	1	3	13	13	3	1	7	7	7
			Machine Architecture								
			1	6	21	21	6	2	9	9	9
Machine Data - Recon Cost 15x			**Machine Configuration**								
Agent	**Material**	**Cost ($/part)**	**Base**	**Table**	**L.S.**	**I.M.**	**Fix.**	**Col.**	**Spin.**	**S.M.**	**Mills**
1	Bronze	2.3	1	1	10	10	1	1	8	8	8
2	100-150 BHN Steel	7.1	1	6	19	19	6	1	7	7	7
3	350-450 BHN Steel	13.8	1	2	14	14	2	1	10	10	10
			Machine Architecture								
			1	6	19	19	6	1	10	10	10

The decreasing trend in machine architecture is shown in Figure 5-15. From this figure, the number of work holding units and columns decrease slightly relative to the change in reconfiguration cost. This slight decrease is the result of the increase in reconfiguration cost. By increasing the reconfiguration cost, the necessity to maintain similar machine architectures between configurations becomes more important. Thus, algorithm forces the architectures into similar states to minimize reconfiguration.

Conversely to the minor change in the number of columns and work holding units, the number of tooling units changes significantly from sixteen to nine. Similar to the work holding units, the changing number of spindles is the result of a changing reconfiguration cost. The changing reconfiguration cost couples with the operating cost which relies upon the number of spindles on the machine to determine the machining cost of the equipment. Both the reconfiguration cost and operating cost benefit from minimizing the size and differences in the machine configurations. Thus resulting in a decreasing part cost trend as observed in the data.

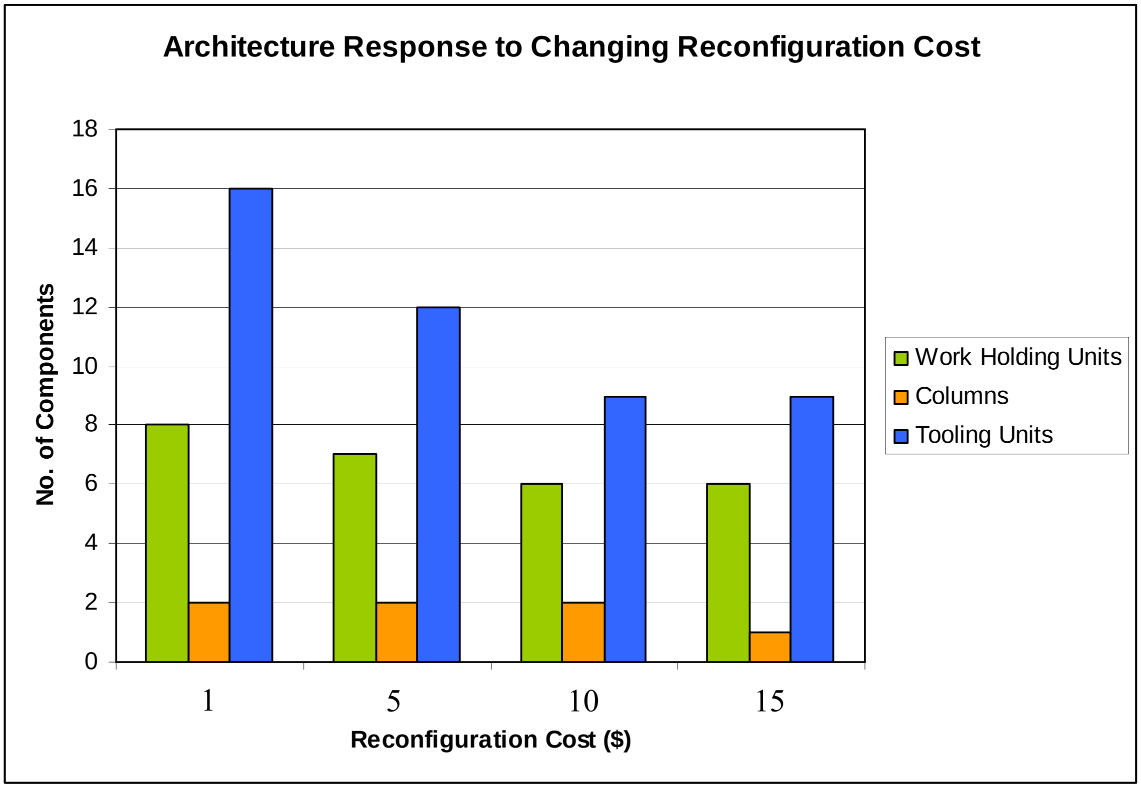


Figure 5-15: Machine architecture evolution relative to changing reconfiguration cost.

To further support the appropriateness of the presented solutions, the convergence plots are shown in Figure 5-16, Figure 5-17, Figure 5-18, and Figure 5-19. In each figure, the agent three average fitness is higher than the agent two average fitness. Agent two average fitness is larger than the agent one average fitness. This behavior is expected due to the material variation and increased operating cost. From this, the solutions are relativistically appropriate.

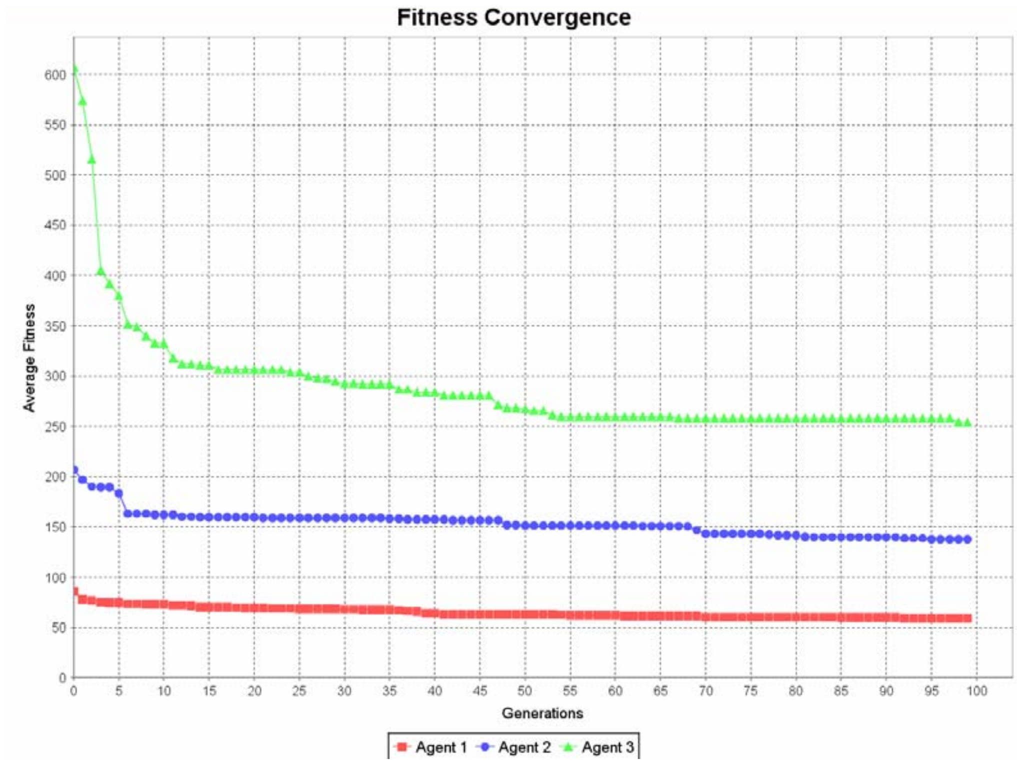


Figure 5-16: Experiment 5 fitness convergence with reconfiguration cost 1x.

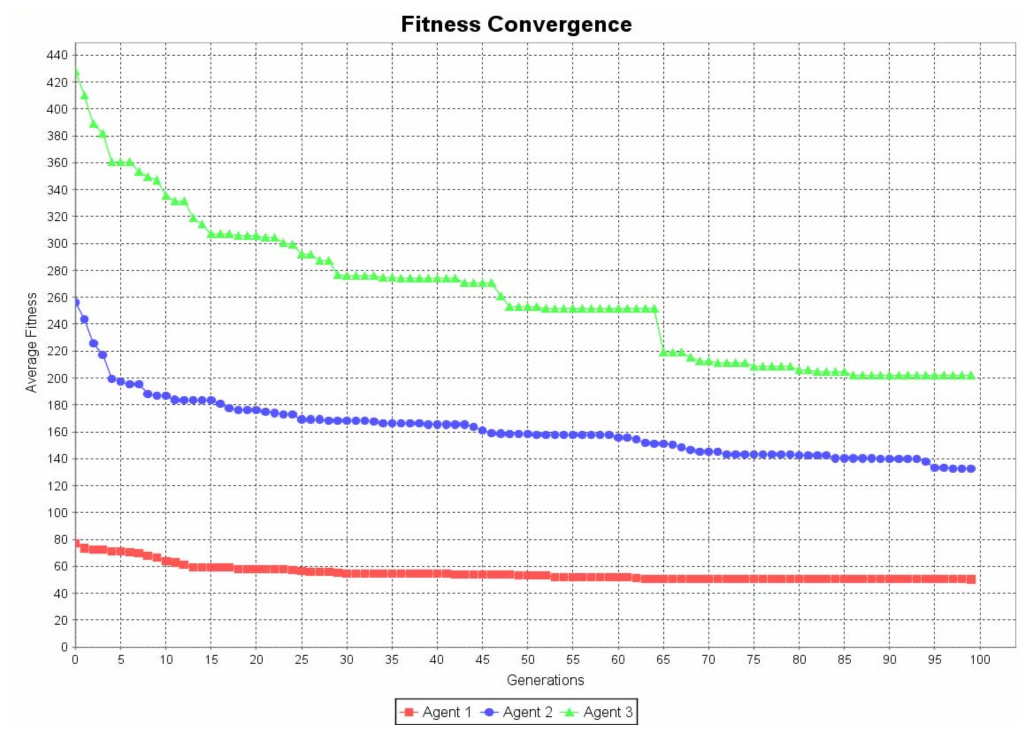


Figure 5-17: Experiment 5 fitness convergence with reconfiguration cost 5x.

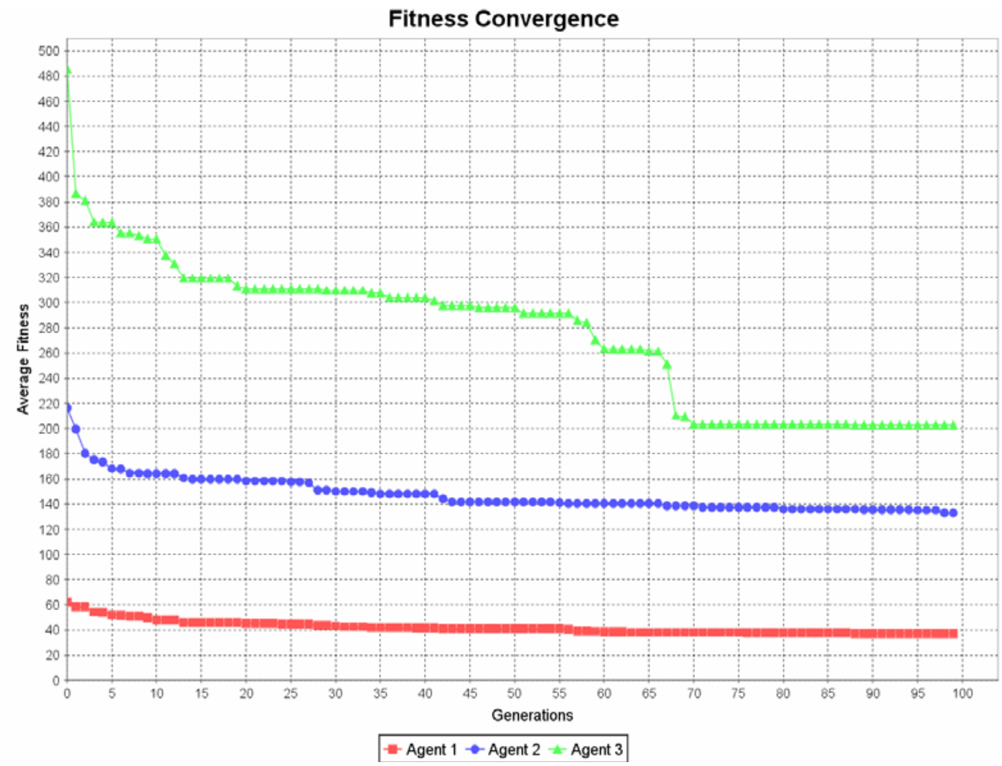


Figure 5-18: Experiment 5 fitness convergence with reconfiguration cost 10x.

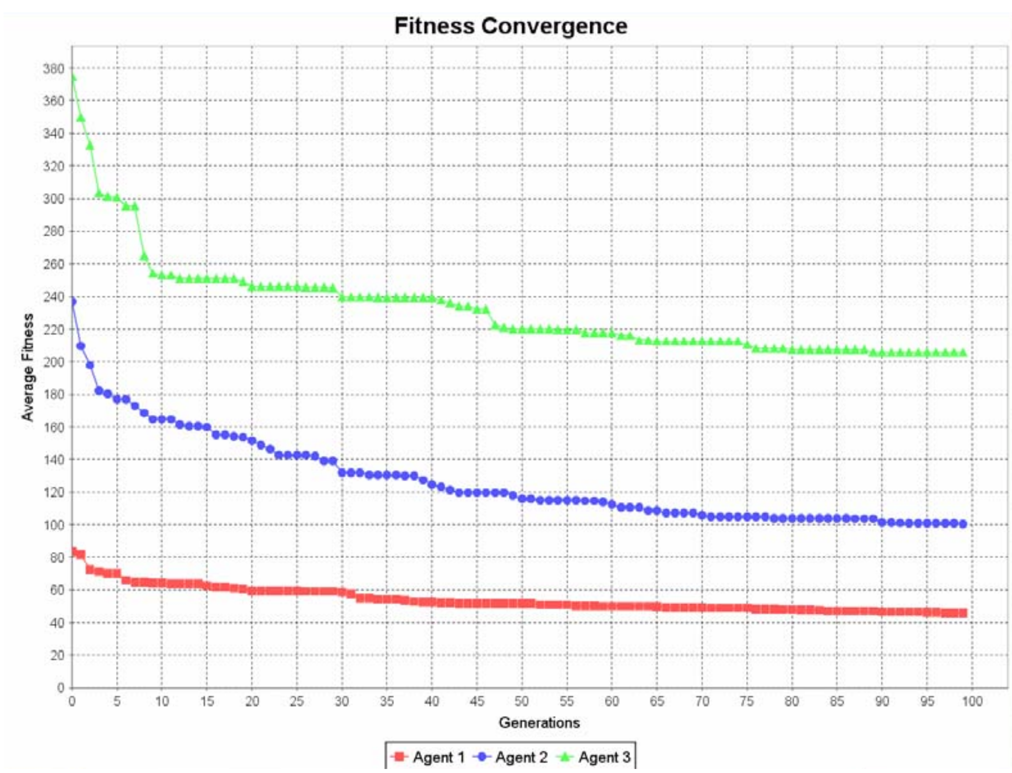


Figure 5-19: Experiment 5 fitness convergence with reconfiguration cost 15x.

5.3 Discussion

To validate the design method with respect to the proposed hypothesis, several experiments were performed which include machine architecture evolution relative to changing batch size, required volume removal, workpiece material, number of features, and reconfiguration cost. These experiments were performed to identify the appropriate architecture for various product ranges.

In experiment one, overall part cost decreases with increasing batch size requirements. This decrease in part cost is the result of identifying progressively fitter machine architectures for the given set of environmental conditions. The results support the proposed hypothesis by providing some evidence of the advantage of employing biological evolution as a design method for evolving manufacturing machine architectures.

Conversely to experiment one, experiment two displays a progressive increase in the overall part cost. This increase is due to the larger volume removal requirements. The machines are required to significantly increase cutting time. In this case, the positive adaptation occurs due to the similarity in the work holding numbers for each machine configuration. The work holding units are more expensive than tooling units in terms of both capital and reconfiguration cost. Thus, the overall number of work holding units and reconfiguration cost are reduced more drastically than tooling units.

Similar to experiment two, experiment three shows both a progressive increase in part cost and a small number of work holding units. The increasing part cost occurs due to increases in unit power and decreases in cutting speed requirements for the material variation. Hence to reduce the cutting time, the machine configurations take on more tooling units to distribute the cutting time amongst the tools and reduce the required batch processing time. Similar to experiment two, the positive adaptation occurs due to the work holding units on each machine. In this case, there is one work holding unit per configuration. This results in zero reconfiguration cost and a minimal amount of capital cost for work holding.

In experiment four, the part cost increases along with the feature number and required material removal as expected. Each machine configuration maintains high numbers of work holding units and tooling units. This behavior is due to the affect of capital cost and reconfiguration cost on the model. Since the affect of capital cost and reconfiguration cost on the part cost is low, the synthesized machines tend to identify architectures which are satisfactory about the initially located machine configurations. The reconfiguration cost does not have a strong affect on reshaping the machine architecture.

In experiment five, the reconfiguration cost can strongly affect the architecture of the machine by increasing the importance of reconfiguration relative to the part cost. The increase of reconfiguration cost results in a mutual benefit between the operating cost and reconfiguration cost resulting in a smaller overall part cost. This behavior occurs due to the operating cost's dependence on the number of machine tooling units. If a machine configuration requires a small number of tooling units to yield a low part cost, then the other machine configurations are driven to a similar small architecture. This reduces the reconfiguration cost due to similarity in machine architecture and reduces the operating cost due to the lower number of spindles.

In conclusion, the experiments present a variety of conditions which are used to validate the machine synthesis algorithm. Experiment one shows a decrease in the part cost with an increase in the batch size. The decrease in part cost is attributed to the progressive development of a more dedicated, concurrent machining configuration. Experiments two and three show an increase in part cost with an increase in volume removal or material hardness. Experiment four shows the small affect of reconfiguration and capital cost on the overall part cost and machine architecture. Experiment five shows the decrease machine architecture size relative to the reconfiguration cost. For each experiment, a representative run of the co-evolutionary algorithm is displayed. For representative runs with a termination criterion of 5000 generations, refer to Appendix C.

In this thesis, a hypothesis is presented stating that biological evolution could inspire a design method for designing the architecture of evolving manufacturing machines. This hypothesis is validated by means of the changing machine architecture in response to changing environmental inputs. In these experiments, the machine configurations adapt to new conditions. In many of these cases, the adaptation is positive relative to the current product range. Thus, biological evolution is shown to provide results which may be useful in the context of designing evolving machines.

Chapter 6 Closure

In this chapter, conclusions are made about the proposed design method and selected experiments to validate the design method. This chapter first discusses the summary of the chapters in Section 6.1. Section 6.2 discusses the answer to the primary research question. In Sections 6.3 and 6.4, a critical evaluation of the method is articulated along with the future work required to satisfy these criticisms.

6.1 Summary of Thesis

This thesis is comprised of five chapters excluding this chapter. In these chapters, a design method is introduced which is inspired from biological adaptation. This design method incorporates co-evolution as a mechanism to design reconfigurable milling machines for adaptation to uncertain product requirement changes. To develop this design method, the necessity for machine level agility is argued in Chapter 1. In this chapter, a need for a design method for reconfigurable manufacturing machine adaptation is developed. to accommodate the necessity for machine agility. To develop a research question for this work, three research areas are reviewed in Chapter 2. These areas include reconfigurable manufacturing systems, reconfigurable robotics, and design methods for reconfigurable systems. From the various approaches in these areas, a research question was posed as follows: *‘What should the architecture of a manufacturing machine be such that it can be reconfigured and adapted to changing needs and opportunities?’* Based on this research question, it is hypothesized that biological evolution could inspire and present an approach designing evolving architectures. Therefore, a design method is proposed in Chapter 3 which is based upon adaptation in natural systems. This design method involves cooperative co-evolution as a mechanism to drive the identification of a series of machine architectures for a changing product range. The details associated with this design approach are presented in Chapter 4. In this design approach, a co-evolutionary multi-agent search algorithm is employed to synthesize machine architectures from a graph based solution representation. Within this algorithm, a concurrent manufacturing simulation is used to evaluate the machine architectures. The results from this algorithm are presented in Chapter 5. Within this chapter, the results are discussed to validate the proposed hypothesis used to answer the research question of this work.

6.2 Answering the Research Question

To answer the research question, the associated hypothesis specifies that biological evolution could be used as inspiration for designing evolving machine architectures. Chapter 3 presents an introduction to co-evolution and an overview of the proposed design method for reconfigurable manufacturing machines. The co-evolutionary method presents a cooperative mechanism which involves the implementation of a resource to resource exchange between machine architectures to facilitate architectural evolution. During this evolution, architectures evolve relative to part cost and architectures for future product demands. Thus, the evolution of the machine architecture is planned in advance. In Chapter 4, the details associated with the development of this type of search algorithm are introduced. The feature which is critical to answering the research question is presented in Section 4.3.6. In this section, the co-evolutionary information is exchanged between machine configurations which update the fitness function upon the identification of a new best solution. Co-adaptation occurs which drives the evolutionary process. The results which validate the proposed design method and research question are presented in Chapter 5. In this chapter, five experiments are presented which display machine evolution relative to changing batch size, volume removal, materials, numbers of features, and reconfiguration cost. These experiments show machine configuration adaptation with respect to various environmental inputs. Furthermore, the adaptations occur in a fashion which follows expected part cost trends with respect to the environmental inputs. Thus, the hypothesis is deemed validated and the research question answered.

6.3 Critical Evaluation of Work

The design method presented in this thesis is shown to be an appropriate method for the synthesis of reconfigurable manufacturing machines. Although the results validate the approach, several limitations are present which go beyond the aforementioned assumptions for the design method and manufacturing model. These limitations include the lack of environmental uncertainty quantification, interference and dynamic cutting models, detailed machine component synthesis, and analysis of network topology.

- This method does not quantify the uncertainty of the product environment to measure extent of machine adaptation. This type of characterization would provide another means of evaluating the quality of machine architecture. In its current state, the design method is validated by showing several experiments and analyzing the data. By implementing a quantification of the product uncertainty, machine architecture could be evaluated by analyzing a specific metric which would account for its ability to accommodate a certain range of uncertainty.

- Also, the simulation does not account for geometric interference of the machine tools or tool path tracing. In the current model, no geometric constraints were considered while synthesizing machines. No information can be given about the machine assembly such as where the column or spindles might be mounted. For instance, concurrent machining may require access to five sides of the workpiece. If the columns are mounted in locations not conducive to this type of machining, the tooling assembly would require significant DOF's to accommodate tool paths. These DOF's might be costly in the context of tooling stiffness and control.
- The focus of this thesis was extending the concepts of biological evolution to the synthesis of manufacturing machines; hence, the level of detail associated with this model pertained solely to the components associated with cutting operations. The current implementation of the model does not contain the capability to deliver specifications such as lead screw thread pitch or the coolant hose line size. With model augmentation, the synthesis algorithm could be used to identify more specific machine component parameters. The constraining factor in this implementation would be the design space size.
- The current assumed network topology could be changed to represent different product ranges. Furthermore, the reconfiguration relationships could have been bi-directional such that agent one had a reconfiguration cost with both agent two and three instead of one relationship with two. Using this type

of topology may have driven the architectures into closer similarity due to the increase in the reconfiguration cost.

6.4 Future Work

From the limitations and criticisms of this design method, a great deal of future work has been identified which could significantly contribute to the development of this approach to designing the architecture of RMMs. This future work may be subdivided into two categories the machine environment and the machine solution representation.

The machine environment represents the inputs which affect the synthesis of suitable machine architectures. In this thesis, the environmental inputs include the product features and batch sizes. To model the uncertainty of the workpiece and batch size, different network states were generated and tested. These states had different products or batch sizes which characterized the possibility new environmental conditions. This approach does not rigorously quantify the uncertainty associated with the changing product geometry. The next research step is to mathematically model the uncertainty associated the product geometry and batch size such that specific metrics can be identified to provide insights into machine adaptation.

Furthermore, the workpiece geometry is characterized using bounding boxes. To better model the product relative to the true required volume removal, tool path tracing could be incorporated such that the exact geometry is represented when machining is simulated. Hence, a further step to better represent the workpiece relates to the incorporation of a higher fidelity machine simulation and representation. In this higher fidelity model, the architecture identified by the current algorithm could be used to synthesize the configuration of the machine with tool path planning and the geometric tooling interference. By identifying the configuration in this way, concurrent machining could be proven relative to the tool path and machine geometry.

Furthermore, a more detailed configuration model would require more machine components. Another research step would include incorporating the synthesis of component specifications such as required motor outputs, lead screw parameters, coolant system locations, and electrical system requirements. Thus, the components required to provide the behavior associated with tool path tracing could be identified.

Finally, several assumptions were made within this work with reference to costs such as operating, reconfiguration, and capital cost. By changing these assumptions the behavior of the synthesis algorithm changes significantly. To synthesize appropriate machine architectures, the assumptions must be correct to enhance the validity of the generated machine architectures. Another research step would be the incorporation of better cost models.

Appendix A

Manufacturing Data

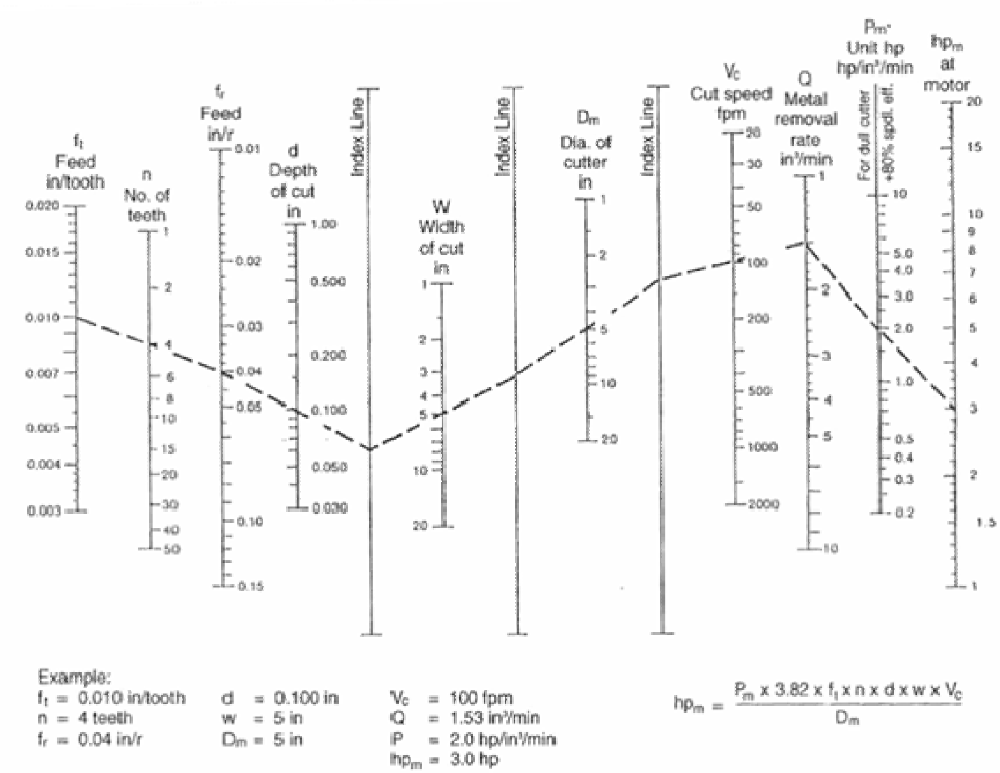


Figure A-1: Alignment chart for metal removal rate [1].

Table A-1: HSS and carbide cutter speeds and feeds per tooth [18].

		Carbide Cutters						High-Speed Steel Cutters					
Material	Feed (in./tooth) Speed (fpm)	Face Mills	Slab Mills	End Mills	Full and Half Side Mills	Saws	Form Mills	Face Mills	Slab Mills	End Mills	Full and Half Side Mills	Saws	Form Mills
Malleable iron	Feed per tooth	.005–.015	.005–.015	.005–.010	.005–.010	.003–.004	.005–.010	.005–.015	.005–.015	.003–.015	.006–.012	.003–.006	.005–.010
Soft/Hard	Speed, fpm	200–300	200–300	200–350	200–300	200–350	175–275	60–100	60–90	60–100	60–100	60–100	60–80
Cast steel	Feed per tooth	.008–.015	.005–.015	.003–.010	.005–.010	.002–.004	.005–.010	.010–.015	.010–.015	.005–.010	.005–.010	.002–.005	.008–.012
Soft/Hard	Speed, fpm	150–350	150–350	150–350	150–350	150–300	150–300	40–60	40–60	40–60	40–60	40–60	40–60
100–150	Feed per tooth	.010–.015	.008–.015	.005–.010	.008–.012	.003–.006	.004–.010	.015–.030	.008–.015	.003–.010	.010–.020	.003–.006	.008–.010
BHN Steel	Speed, fpm	450–800	450–600	450–600	450–800	350–600	350–600	80–130	80–130	80–140	80–130	70–100	70–100
150–250	Feed per tooth	.010–.015	.008–.015	.005–.010	.007–.012	.003–.006	.004–.010	.010–.020	.008–.015	.003–.010	.010–.015	.003–.006	.006–.010
BHN Steel	Speed, fpm	300–450	300–450	300–450	300–450	300–450	300–450	50–70	50–70	60–80	50–70	50–70	50–70
250–350	Feed per tooth	.008–.015	.007–.012	.005–.010	.005–.012	.002–.005	.003–.008	.005–.010	.005–.010	.003–.010	.005–.010	.002–.005	.005–.010
BHN Steel	Speed, fpm	180–300	150–300	150–300	160–300	150–300	150–300	35–60	35–50	40–60	35–50	35–50	35–50
350–450	Feed per tooth	.008–.015	.007–.012	.004–.008	.005–.012	.001–.004	.003–.008	.003–.008	.005–.008	.003–.101	.003–.008	.001–.004	.003–.008
BHN Steel	Speed, fpm	125–180	100–150	100–150	125–180	100–150	100–150	20–35	20–35	20–40	20–35	20–35	20–35
Cast Iron Hard	Feed per tooth	.005–.010	.005–.010	.003–.008	.003–.010	.002–.003	.005–.010	.005–.012	.005–.010	.003–.008	.005–.010	.002–.004	.005–.010
BHN 180–225	Speed, fpm	125–200	100–175	125–200	125–200	125–200	100–175	40–60	35–50	40–60	40–60	35–60	35–50
Cast Iron Medium	Feed per tooth	.008–.015	.008–.015	.005–.010	.005–.012	.003–.004	.006–.012	.010–.020	.008–.015	.003–.010	.008–.015	.003–.005	.008–.012
BHN 180–225	Speed, fpm	200–275	175–250	200–275	200–275	200–250	175–250	60–80	50–70	60–90	60–80	60–70	50–60
Cast Iron Soft	Feed per tooth	.015–.025	.010–.020	.005–.012	.008–.015	.003–.004	.008–.015	.015–.030	.010–.025	.004–.010	.010–.020	.002–.005	.010–.015
BHN 150–180	Speed, fpm	275–400	250–350	275–400	275–400	250–350	250–350	80–120	70–110	80–120	80–120	70–110	60–80
Bronze	Feed per tooth	.010–.020	.010–.020	.005–.010	.008–.012	.003–.004	.008–.015	.010–.025	.008–.020	.003–.010	.008–.015	.003–.005	.008–.015
Soft/Hard	Speed, fpm	300–100	300–800	300–1000	300–1000	300–1000	200–800	50–225	50–200	50–250	50–225	50–250	50–200
Brass	Feed per tooth	.010–.020	.010–.020	.005–.010	.008–.012	.003–.004	.008–.015	.010–.025	.008–.020	.005–.015	.008–.015	.003–.005	.008–.015
Soft/Hard	Speed, fpm	500–1500	500–1500	500–1500	500–1500	500–1500	500–1500	150–300	100–300	150–350	150–350	150–300	100–300
Aluminum Alloy	Feed per tooth	.010–040	.010–.030	.003–.015	.008–.025	.003–.006	.008–.015	.010–.040	.015–.040	.015–.040	.010–.030	.004–.008	.010–.020
Soft/Hard	Speed, fpm	2000 UP	2000 UP	2000 UP	2000 UP	2000 UP	2000 UP	300–1200	300–1200	300–1200	300–1200	300–1000	300–1200

*Generally, lower end of range used for inserted blade cutters, higher end of range for indexable insert cutters.

Table A-2: Unit power information [18].

Material (Hardness)	Unit Power (hp-min./in^3) HP_s	Specific Energy (in.-lb/in^3) K_s or U
Steel (120 Bhn)	1.12	443,000
Steel (120 Bhn)	0.86	347,000
Steel (120 Bhn)	0.76	301,000
Steel (120 Bhn)	0.64	254,000
Steel (120 Bhn)	0.54	214,000
Steel (160 Bhn)	1.25	495,000
Steel (160 Bhn)	0.59	234,000
Steel (200 Bhn)	1.50	594,000
Steel (200 Bhn)	0.73	290,000
Steel (300 Bhn)	1.87	740,000
Steel (300 Bhn)	0.92	364,000
SAE-302	0.72	285,000
SAE-350	1.20	475,000
SAE-410	0.75	297,000
Gray CI (130 Bhn)	0.29–0.33	127,000
Meehanite	0.55–0.76	262,000
K-Monel	0.80	317,000
Inconel 700	1.40	554,000
High-Temperature Alloy A 286	1.20	475,000
High-Temperature Alloy S 816	1.25	495,000
Titanium A-55	0.65–0.76	281,000
Titanium C-130	0.81–0.93	345,000
Titanium (250–275 BHN)	1.8–2.0	
Aluminum 2014-T6, 2014-T4	0.24	95,100
Aluminum 6064-TO	0.34	125,000
Aluminum 3003-O	0.16	63,400
Aluminum 108 (55 BHN)	0.15	49,400
Muntz Metal	0.55	218,000
Phosphor Bronze	0.33	131,000
Cartridge Brass	0.48	190,000
Copper Alloys (10–80 R_B)	0.5–0.6	
Copper (50 R_B)	0.9–1.0	
Magnesium (40–90 BHN at 500 kg)	0.16	
Tungsten, Tantilum (210–320 BHN)	2.6–2.8	
Nickel Alloys (280–360 BHN)	1.8–2.0	
Nickel/Cobalt Alloys (200–360 BHN)	2.0–2.5	

Values assume normal feed ranges and sharp tools. Multiply values by 1.25 for a dull tool.

Calculation of Unit Power (HP_s)

$$HP = F_c V/33000$$

HP_s = HP/MRR Where

MRR = $12Vtw$ for tube turning

$$HP_s = F_c V/12Vtw \times 33000 = F_c/tw \times 396000$$

Calculation of specific energy (U)

$U = F_c V/Vtw = F_c/tw$ for tube turning

Appendix B

Repository of Assumptions

The following list represents a repository of the assumptions made during the development of this synthesis algorithm.

- The spindle drive motor power is 5 hp [18].
- The spindle drive motor efficiency is 0.8 [18].
- The operating cost is assumed to be one dollar per minute per spindle [18].
- One half of a minute is required to load and unload parts in the fixture [18].
- Nonproductive cost per piece is one half of a dollar [18].
- Tool life is neglected in the machine synthesis algorithm.
- Tooling and tool changing cost are not incorporated in the fitness function.
- Horizontal milling machine arrangements are not included; therefore, only vertical mill components are included.
- Only components specific to the cutting simulation are included in the model.
- Detailed components such as material transfer equipment and sensors are not included.

- Subsystems not directly associated with the cutting equations such as hydraulics, pneumatics, coolant, or electrical systems are not considered even though their presence is required for proper machine operation.
- The single spindle cost model can be extended to multiple spindles working concurrently on the same workpiece.
- Reconfiguration cost is assumed to be an accurate model of measuring the conversion of one machine configuration to the next configuration.
- All spindles operate concurrently.
- Speed of machine is determined by the slowest cutting tool.
- Geometric machine interference details are ignored.
- Tool path considerations are ignored.
- Features are assumed to be represented by a bounding box containing height, width, and depth.

Appendix C

Extra Representative Runs

Table C-1: Experiment 1 5000 generation machine architecture.

Machine Data			Machine Configuration								
Agent	Batch Size	Cost ($/part)	Base	Table	L.S.	I.M.	Fix.	Col.	Spin.	S.M.	Mills
1	5000	5.7	1	1	12	12	1	1	10	10	10
2	10000	7.8	1	1	23	23	1	2	21	21	21
3	15000	8.1	1	1	11	11	1	1	9	9	9
			Machine Architecture								
			1	1	23	23	1	2	21	21	21

Figure C-1: Experiment 1 5000 generation fitness convergence.

Table C-2: Experiment 2 5000 generation machine architecture.

Machine Data			Machine Configuration								
Agent	Volume (cu.in.)	Cost ($/part)	Base	Table	L.S.	I.M.	Fix.	Col.	Spin.	S.M.	Mills
1	6.57	1.3	1	1	9	9	1	1	7	7	7
2	52.59	8.7	1	1	11	11	1	1	9	9	9
3	177.49	19.8	1	1	11	11	1	1	9	9	9
			Machine Architecture								
			1	1	11	11	1	1	9	9	9

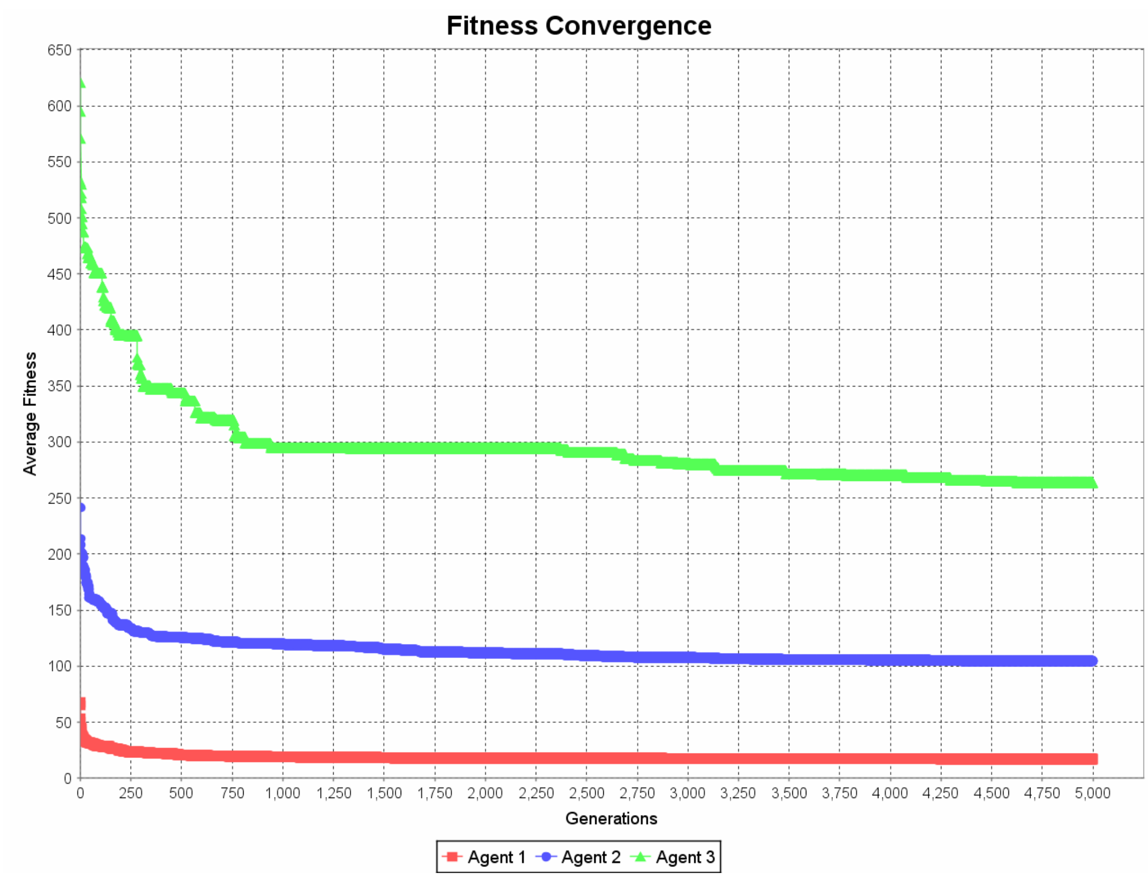


Figure C-2: Experiment 2 5000 generation fitness convergence.

Table C-3: Experiment 3 5000 generation fitness convergence.

Machine Data			Machine Configuration								
Agent	Material	Cost ($/part)	Base	Table	L.S.	I.M.	Fix.	Col.	Spin.	S.M.	Mills
1	Bronze	9.4	1	1	16	16	1	1	14	14	14
2	100-150 BHN Steel	21.9	1	1	9	9	1	1	7	7	7
3	350-450 BHN Steel	62.3	1	1	11	11	1	1	9	9	9
			Machine Architecture								
			1	1	16	16	1	1	14	14	14

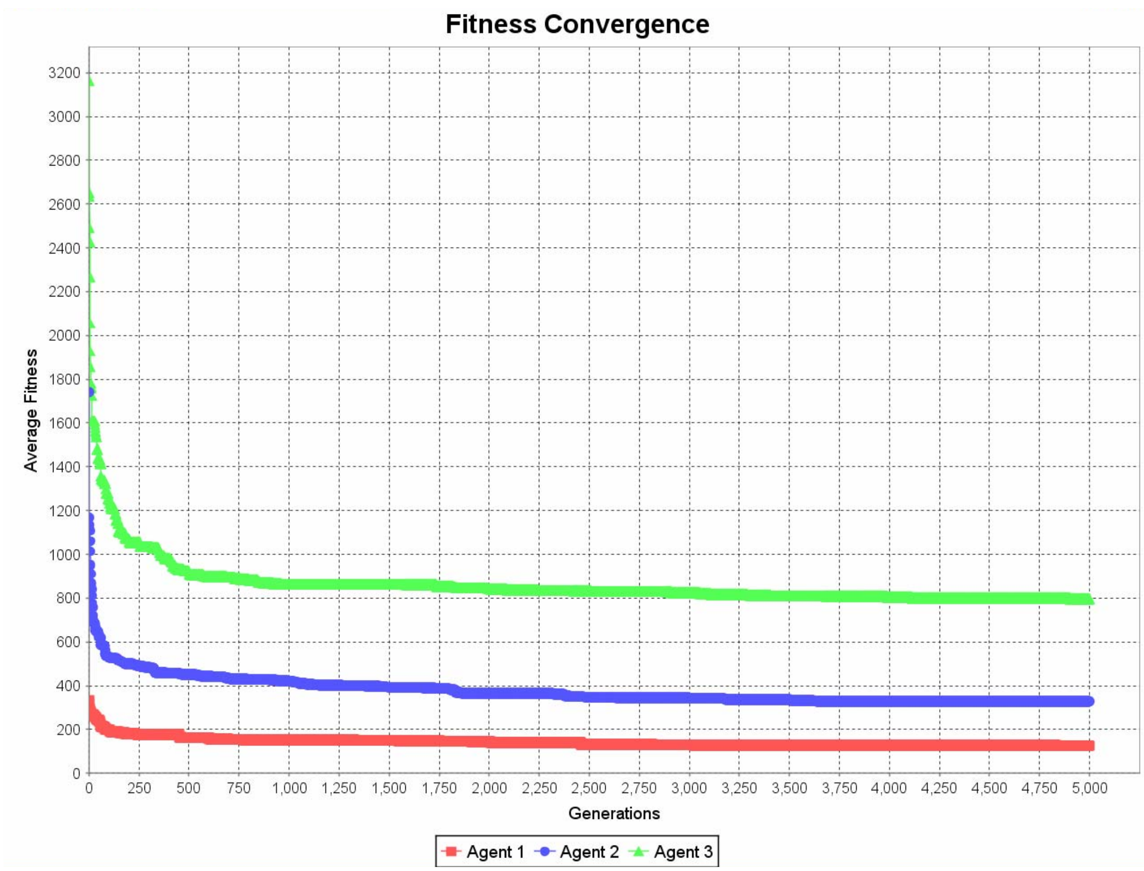


Figure C-3: Experiment 3 5000 generation fitness convergence.

Table C-4: Experiment 4 5000 generation machine architecture.

Machine Data			Machine Configuration								
Agent	Features	Cost ($/part)	Base	Table	L.S.	I.M.	Fix.	Col.	Spin.	S.M.	Mills
1	6.57	8.6	1	1	10	10	1	1	8	8	8
2	52.59	9.1	1	1	12	12	1	1	10	10	10
3	177.49	11.5	1	1	11	11	1	1	9	9	9
			Machine Architecture								
			1	1	12	12	1	1	10	10	10

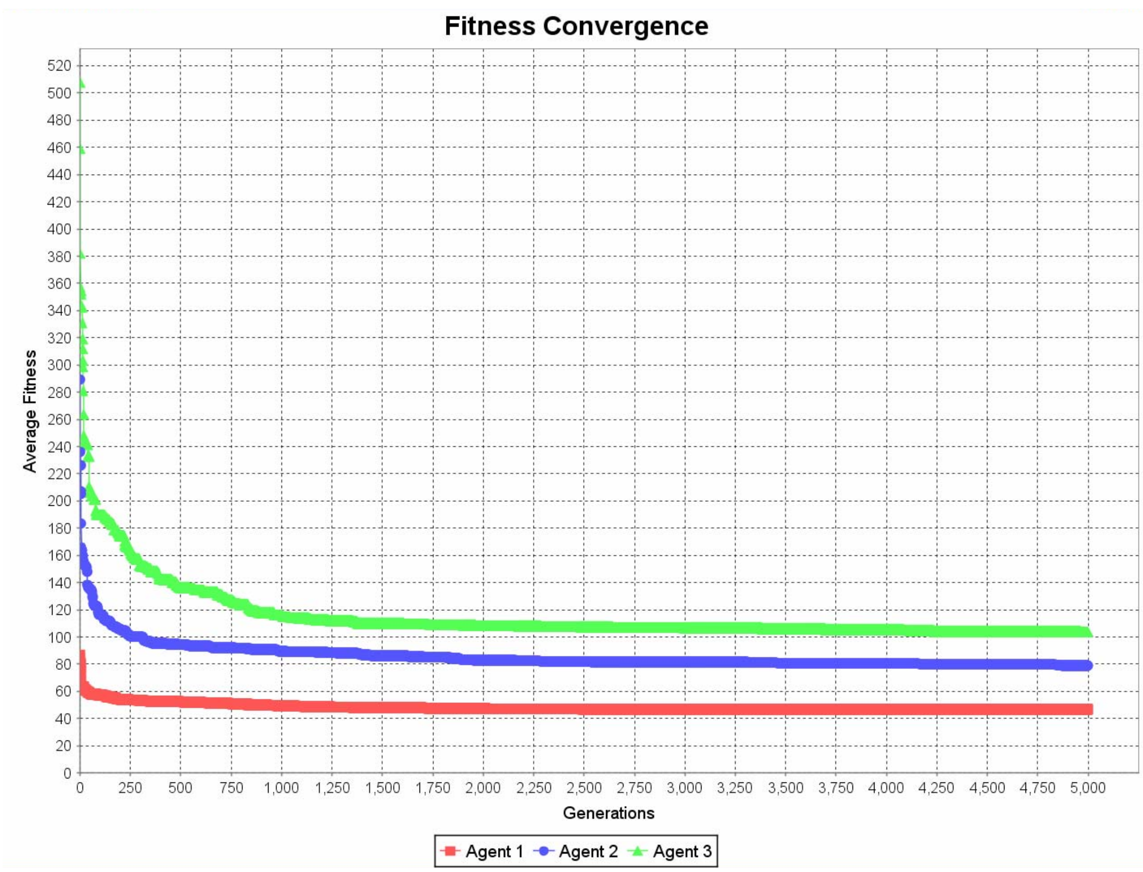


Figure C-4: Experiment 4 5000 generation fitness convergence.

References

[1] *Mechanical Engineers' Handbook*. 2nd ed, ed. M. Kutz. 1998, New York, NY: John Wiley & Sons, Inc. 2352.

[2] *ABB Launches New Robot for Cleaning of Aluminum Castings*. 2002, Available from: http://www.abb.com/, [cited 2008 15 February].

[3] *Machining with Kuka and Delcam Flyer*. 2005 Available from: http://www.roboticmachining.com/, [cited 2008 15 February].

[4] *Warco Super WM16 CNC Milling Machine*. 2006, Available from: http://www.simplycnc.com/, [cited 2008 20 February].

[5] Axelrod, R., *The Evolution of Cooperation*. 1984, New York, NY: Basic Books. 1-231.

[6] Backhed, F., Ley, R.E., Sonnenburg, J.L., Perterson, D.A., Gordon, J.I., *Host-Bacterial Mutualism in the Human Intestine.* Science, 2005. **307**: p. 1915 - 1920.

[7] Bever, J.D. *Negative Feedback within a Mutualism: Host-Specific Growth of Mycorrhizal Fungi Reduces Plant Benefit*. in *Proceedings of Royal Society of London B*. 2002. London, England.

[8] Bi, Z.M., Lang, S.Y.T., Shen, W., Wang, L., *Reconfigurable Manufacturing Systems: The State of the Art.* International Journal of Production Research, 2007. **46**(4): p. 967-992.

[9] Bollinger, J.G., Benson, D.K., Cloud, N., *Visionary Manufacturing Challenges for 2020*, in *National Research Council Report*. 1998, National Research Council: Washington, D.C.

[10] Bronstein, J.L., *The Contribution of Ant-Plant Protection Studies to Our Understanding of Mutualism.* Biotropica, 1998. **30**(2): p. 150-161.

[11] contributors, W. *Cleaner fish* 2008 15 November 2007, Available from: http://en.wikipedia.org/, [cited 2008 10 February].

[12] contributors, W. *Co-evolution*. 2008 29 January 2008, Available from: http://en.wikipedia.org/, [cited 2008 10 February].

[13] contributors, W. *Evolution*. 2008 18 February, 2008, Available from: http://en.wikipedia.org/, [cited 2008 20 February].

[14] contributors, W. *Mutualism* 2008 5 February 2008, Available from: http://en.wikipedia.org/, [cited 2008 10 February].

[15] Darwin, C., *The Origin of Species*. 1859, London, England: John Murray.

[16] Dawkins, R., Krebs, J.R., *Arms Races Between and Within Species.* Proceedings of Royal Society of London B, 1979. **205**: p. 489-511.

[17] de Weck, O.L., Nadir, W.D., Wong, J.G., Bounova, G., Coffee, T.M. *Modular Structures for Manned Space Exploration: The Truncated Octahedron as a Building Block*. in *1st Space Exploration Conference: Continuing the Voyage of Discovery*. 2005. Orlando, FL: AIAA.

[18] Degarmo, E.P., Black, J.T., Kohser, R.A., *Manufacturing Automation*, in *Materials and Processes in Manufacturing*, J. Hayton, Editor. 2003, John Wiley & Sons, Inc.: Danvers, MA.

[19] Duff, D.G., Yim, M., Roufas, K. *Evolution of PolyBot: A Modular Reconfigurable Robot*. in *Proceedings of the Harmonic Drive International Symposium*. 2001. Nagano, Japan.

[20] Ferguson, S.M., Lewis, K., *Effective Development of Reconfigurable Systems Using Linear State-Feedback Control.* AIAA, 2006. **44**(4): p. 868-878.

[21] Futuyma, D.J., *Evolution*. 2005, Sunderland, MA: Sinauer Associates, Inc. 543.

[22] Heisel, U., Meitzner, M., *Mobile Reconfigurable Manufacturing Unit for Drilling and Milling Operations.* International Journal of Flexilbe Manufacturing Systems, 2006. **17**: p. 315-322.

[23] Holland, J.H., *Adaptation in Natural and Artificial Systems*. 1975, Ann Arbor, MI: University of Michigan Press. 7.

[24] Holland, J.H., *Hidden Order How Adaptation Builds Complexity*. 1995, New York, NY: Basic Books. 185.

[25] Kiers, E.T., Rousseau, R.A., West, S.A., Denison, R.F., *Host Sanctions and the Legume-Rhizobium Mutualism.* Nature, 2003. **425**: p. 78-81.

[26] Koren, Y., Hu, J., Weber, T.W., *Impact of Manufacturing System Configuration on Performance.* Annals of the CIRP, 1998. **47**(1): p. 369-372.

[27] Koren, Y., Katz, R., *Reconfigurable Apparatus and Method for Inspection During a Manufacturing Process*, U.S.P. Office, Editor. 2003, The Regents of the University of Michigan: United States. p. 1-6.

[28] Koren, Y., Kota, S., *Reconfigurable Machine Tool*, U.S.P. Office, Editor. 1999, The Regents of the University of Michigan, Ann Arbor, Mich.: United States. p. 1-7.

[29] Koren, Y., Ulsoy, A.G., *Reconfigurable Manufacturing System Having a Production Capacity Method for Designing Same and Method for Changing its Production Capacity*, U.S.P. Office, Editor. 2002, The Regents of the University of Michigan: United States. p. 1-12.

[30] Koren, Y.H., U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G., Brussel, H.V., *Reconfigurable Manufacturing Systems.* Annals of the CIRP, 1999. **48**(2): p. 527-540.

[31] Kurokawa, H., Kamimura, A., Yoshida, E., Tomita, K., Kokaji, S., Murata, S. *M-TRAN II: Metamorphosis from a Four-Legged Walker to a Caterpillar*. in *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2003. Las Vegas, Nevada.

[32] Landers, R., Min, B.K., Koren, Y., *Reconfigurable Machine Tools.* Annals of the CIRP, 2001. **50**(1): p. 269-274.

[33] Maier-Speredelezzi, V., Koren, Y., Hu, S.J., *Convertibility Measure for Manufacturing Systems.* CIRP Annals of Manufacturing Technologies, 2003. **52**(1): p. 367-370.

[34] Mehrabi, M.G., Ulsoy, A.G., Koren, Y., *Reconfigurable Manufacturing Systems: Key to Future Manufacturing.* Journal of Intelligent Manufacturing, 2000. **11**: p. 403-419.

[35] Moon, Y.-M., Kota, S., *Design of Reconfigurable Machine Tools.* Journal of Manufacturing Science and Engineering, 2002. **124**: p. 480-483.

[36] Morey, B., *Robotic Material Removal and Machining*, in *Robotics*. 2007, Society of Manufacturing Engineers.

[37] Murata, S., Yoshida, E., Kamimura, A., Kurokawa, H., Tomita, K., Kokaji, S., *M-TRAN: Self-Reconfigurable Modular Robotic System.* IEEE/ASME Transactions on Mechatronics, 2002. **7**(4): p. 431-441.

[38] Mytilinaios, E., Desnoyer, M., Marcus, D., Lipson, H. *Designed and Evolved Blueprints for Physical Self-Replicating Machines*. in *Proceedings of the 9th International Conference on the Simulation and Synthesis of Living Systems (Artificial Life IX)*. 2004. MIT, Cambridge, MA.

[39] Nadir, W.D., Kim, I.Y., Hauser, D., de Weck, O.L., *Multidisciplinary Structural Truss Topology Optimization for Reconfigurability*, in *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. 2004: Albany, New York. p. 1-15.

[40] Olewnik, A., Brauen, T., Ferguson, S., Lewis, K., *A Framework for Flexible Systems and its Implementation in Multiattribute Decision Making*, in *ASME 2001 Design Engineering Technical Conferences and Computers and Information Engineering Conference*. 2001: Pittsburgh, PA. p. 1-10.

[41] Ollerton, J., McCollin, D., Fautin, D.G., Allen, G.R. *Finding NEMO: Nestedness Engendered by Mutualistic Organization in Anemonefish and their Hosts*. in

Proceedings of Royal Society of London B. 2006. London, England: The Royal Society.

[42] Saaty, T.L., *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*. 1980, New York, NY: McGraw-Hill.

[43] Salemi, B., Moll, M., Shen, W.M. *SUPERBOT: A Deployable, Multi-Functional and Modular Self-Reconfigurable Robotic System*. in *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2006. Beijing, China: IEEE.

[44] Shen, W.-M., Krivokon, M., Chiu, H., Everist, J., Rubenstein, M., Venkatesh, J. *Multimode Locomotion via SuperBot Robots*. in *Proceedings of the 2006 IEEE Conference on Robotics and Automation*. 2006. Orlando, FL: IEEE.

[45] Siddiqi, A., de Weck, O.L. , *Spare Parts Requirements for Space Missions with Reconfigurability and Commonality.* Journal of Spacecraft and Rockets, 2007. **44**(1): p. 147-156.

[46] Siddiqi, A., de Weck, O.L., Iagnemma, K., *Reconfigurability in Planetary Surface Vehicles: Modeling Approaches and Case Study.* J Br Interplanetary Society (JBIS), 2006. **59**: p. 450-460.

[47] Spicer, P., Yip-Hoi, D., Koren, Y., *Scalable Reconfigurable Equipment Design Principles.* International Journal of Production Research, 2005. **43**(22): p. 4839-4852.

[48] van Valen, L., *A New Evolutionary Law.* Evolutionary Theory, 1973. **1**: p. 1-30.

[49] Yim, M., Duff, D.G., Roufas, K.D. *PolyBot: A Modular Reconfigurable Robot*. in *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*. 2000. San Francisco, CA: IEEE.

[50] Yim, M., Roufas, K., Duff, D., Zhang, Y., Eldershaw, C., Homans, S., *Modular Reconfigurable Robots in Space Applications.* Autonomous Robots, 2003. **14**: p. 225-237.

[51] Yim, M., Shen, W., Salemi, B., Rus, D., Moll, M., Lipson, H., Klavins, E., Chirikjian, G.S., *Modular Self-Reconfigurable Robot Systems*, in *IEEE Robotics and Automation Magazine*. 2007, IEEE. p. 1-11.

[52] Yim, M., Zhang, Y., Duff, D., *Modular Robots*, in *IEEE Spectrum*. 2002, IEEE. p. 30-34.

[53] Yim, M., Zhang, Y., Roufas, K., Duff, D., Eldershaw, C., *Connecting and Disconnecting for Chain Self-Reconfiguration with PolyBot.* IEEE/ASME Transactions on Mechatronics, 2002. **7**(4): p. 442-451.

[54] Yoshida, E., Murata, S., Kamimura, A., Tomita, K., Kurokawa, H., Kokaji, S. *Evolutionary Synthesis of Dynamic Motion and Reconfiguration Process for a Modular Robot M-TRAN*. in *Proceedings 2003 IEEE International Symposium on Comutational Intelligence in Robotics and Automation*. 2003. Kobe, Japan: IEEE.

[55] Youssef, A.M.A., ElMaragy, H.A., *Assessment of Manufacturing Systems Reconfiguration Smoothness.* International Journal of Manufacturing Technology, 2006. **30**: p. 174-193.

[56] Zykov, V., Mytilinaios, E., Adams, B., Lipson, H., *Self-Reproducing Machines*, in *Nature*. 2005, Nature Publishing Group. p. 163-164.